Analysis and Control of Mobile Robots in Various Environmental Conditions

A thesis submitted towards partial fulfilment of
The requirements for the degree of

Master of Technology (Research) in
Industrial Design

By

Supriya Mahapatra

610ID301

Department of Industrial Design
National Institute of Technology, Rourkela
Odisha (India)-769008

Analysis and Control of Mobile Robots in Various Environmental Conditions

A thesis submitted towards partial fulfilment of
The requirements for the degree of

Master of Technology (Research) in
Industrial Design

By

Supriya Mahapatra

610ID301

Under the joint Supervision of

Prof. D. R. Parhi
&
Prof. J.Srinivas

Department of Industrial Design
National Institute of Technology, Rourkela
Odisha (India)-769008

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Supriya Mahapatra

Date: **Supriya Mahapatra**

National Institute of Technology, Rourkela
Certificate

This is to certify that the project entitled, "**Analysis and Control of Mobile Robots in Various Environmental Conditions"** submitted by **Miss Supriya Mahapatra** is an authentic work carried out by her under my supervision and guidance for the partial fulfilment of the requirements for the award of **Master of Technology Degree (Research)** in **Industrial Design** at **National Institute of Technology, Rourkela**.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University or Institute for the award of any Degree or Diploma.

Date
Rourkela

Prof. J.Srinivas
(Co. Supervisor)
Dept. of Mechanical Engineering

Prof. D. R. Parhi
(Supervisor)
Dept. of Mechanical Engineering

Acknowledgement

I express my profound gratitude and indebtedness to **Prof. D.R. Parhi**, and **Prof. J.Srinivas** Department Of Mechanical Engineering, NIT, Rourkela for introducing the present topic and for their inspiring intellectual guidance, constructive criticism and valuable suggestion throughout the project work.

I express my sincere thanks to **Prof. B.B. Biswal**, HOD and also the other staff members of Department of Industrial Design, NIT Rourkela for providing me the necessary facilities that is required to conduct the experiment and complete my thesis.

I wish to extend special thanks to **Miss. Subhasree Kundu**, **Mr. Alok Jha**, **Mr. Chinmaya Sahu** and other staffs of Robotics Lab for their valuable suggestions and cordial environment in NIT Campus, throughout my entire project work.

I am deeply indebted to my parents who always believed in me and gave all their support for all the choices that I have made.

Finally, I humbly bow my head with utmost gratitude before the God Almighty who always showed me a path to go and without whom I could not have done any of these.

Supriya Mahapatra

Date
Rourkela **Supriya Mahapatra**

ABSTRACT

The world sees new inventions each day, made to make the lifestyle of humans more easy and luxurious. In such global scenario, the robots have proved themselves to be an invention of great importance. The robots are being used in almost each and every field of the human world. Continuous studies are being done on them to make them simpler and easier to work with. All fields are being unraveled to make them work better in the human world without human interference. We focus on the navigation field of these mobile robots. The aim of this thesis is to find the controller that produces the most optimal path for the robot to reach its destination without colliding or damaging itself or the environment. The techniques like Fuzzy logic, Type 2 fuzzy logic, Neural networks and Artificial bee colony have been discussed and experimented to find the best controller that could find the most optimal path for the robot to reach its goal position. Simulation and Experiments have been done alike to find out the optimal path for the robot.

INDEX

LIST OF FIGURES

SERIAL NUMBERS **PAGE**

LIST OF TABLES

SERIAL NUMBERS **PAGE**

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

The thesis conveys an inventive background, for enabling the mobile robot to explore a congested and cluttered real world surrounding safely, especially, an impulsively fluctuating environment thereby avoiding structured or unstructured obstacles. The work described in this thesis has been carried out in the context of the navigation through various environments with mobile robots. This chapter specifies background information and the basic concept and an overview of the research areas concerning the work carried out as well as the motivation pertaining to the work carried out in this thesis. It then briefly enlightens the overview of major goals of this research i.e. what type of demanding problems have been undertaken and how, which are reaffirmed later in more depth in the successive thesis chapters. Finally, the thesis structure has been sketched.

1.1 BACKGROUND AND NAVIGATION

Starting from the earliest to the latest, the formation of autonomous mobile robots has been acknowledged. For the purpose of efficient control of a robot substantial variations in the robot‛s inclusive behaviour or action are needed in order to behave in large scale surroundings. Mobile robot is not only a combination of algorithms for sensing real time response but also expanding possession of knowledge, rationalizing the positional error and moving about space. It also leads to physical incarnations of these algorithms and ideas, which are able to conduct all of whims of the real world and are also entailed to be coupled. As such mobile robot provides an authenticity check for hypothetical concepts and algorithms.

Research and development of mobile robots have attracted the attention of researchers in the areas of engineering, computer science, biology, mining and others. Mobile robots have a high potential in several applications which include automatic freeway driving, guidance of the blind and disabled, explorations of dangerous regions and mechanical parts transfer in flexible assembly system. Progress in the field of mobile robot navigation has been slower than expectations from the excitement and relatively rapid advances of the early days of research.

A robot acting independently in complicated surroundings has been proven only in very limited number of trials. Autonomous mobile robots are intelligent agents that can perform desired tasks in various (known and unknown) environments without continuous human interference. All kinds of robots are autonomous to some degree. One important area of robotics research is to enable the robot to cope with its environment whether on land, underwater, in the air, underground or in space. A fully autonomous robot in the real world has the ability to

- ❖ Track information from the environment
- ❖ Navigate from one point to another without manual help
- ❖ Avoid obstacles and hazardous situations
- ❖ Repair itself to some extent without outside assistance

A perceptive robot needs to be able to deal with probable, equivocal, inconsistent and noisy data by learning through its own interface with the world while achieving goal. Mechanisms, used in successful navigation of robotic agent, use a number of skills, starting from high level capabilities such as surveying the surrounding environment, to building an autonomous global map and planning a path towards an explicit goal, to the execution of rudimentary low level action like avoiding collisions with obstacles. So an autonomous robot needs to learn the following:

- ❖ Gaining some new abilities
- ❖ Planning navigation strategies based on surroundings
- ❖ Adapting to surroundings without interference

The navigation and control of mobile robots is a challenging topic and can be taken up for research for the following reasons:

- ❖ A mobile robot needs to sense its own environment and move about in the environment without looking forward to human assistance. It should be able to identify different features, detect various obstacles, find paths around the obstacles and navigate accordingly.

- ❖ Autonomous mobile robots are supposed to be the nearest approximations of intelligent agents which means that they satisfy the need of the people to make machines that can mimic living beings or natural phenomena.

- ❖ The applications of mobile robots lead them to operate in environments where even human intrusion is not possible. Hence the autonomous vehicle has to be able enough to navigate in such spaces and endure such circumstances which living beings are unable to tolerate. However for such mobile robots the risk of hurting someone or something is at its least as compared to the robots which have to share their working space with the human beings. In such cases, they have to move within an environment full of uncertain human intrusion and hence the risk of hazardous accidents increases.

In order to mix and adapt to their environments, animals approximate the result of their actions and acquire or adapt their behaviours according to the behaviour of other objects. Hence there is a strong stifle in order to investigate intelligent behaviour by means of positioned agents or mobile robots. Perception and action are imposed to be highly coupled in a closed loop to spawn navigational strategy of mobile agents. This awareness reverses the inclination of mobile robotics field towards an inherently interdisciplinary research area involving the followings:

- For designing locomotive mechanisms : Mechanical Engineering
- For sensing and path planning : Computer Science engineering
- For system integration and communication : Electrical engineering
- For perceptions and neurological behaviour : Cognitive psychology

Path analysis and planning is an exciting challenge in building autonomous mobile robots. An autonomous mobile robot should be able to learn and sense its environment thereby programming itself without assistance. It consists of finding a route from the origin of the robot to its target destination. Path analysis and planning becomes more difficult when some static as well as dynamic obstacles are added to the environment. In such case, it is necessary to find an alternative route. This requires a process of adaptation to the environment. In addition to avoiding collision, the other requirements for navigation in such cases include smoother motion, shorter travelling time, or more clearance from the obstacle. Therefore, the path analysis and planning involves optimization with respect to certain performance measures.

1.2 OBJECTIVE AND GOAL

The goal of autonomous mobile robotics is to build and control physical systems which can move purposefully and without human intervention in real-world environments. To survive within unforeseen situations and to adapt to the effects of changing environment; the power of self-government or sturdy autonomy is essential, which implies that the robot should be able to govern its course of action by its own perceptive process, rather than following a fixed, hardwired sequence of superficially provided instructions. The development of techniques for autonomous mobile robot operation constitutes one of the major trends in the current research and practice in modern robotics.

The objective of this research is to determine the shortest path from the origin of the robot to its target destination. Methods for finding the shortest path, by diverting through obstacles, have traditionally been based on one of several models. This thesis proposes alternative ways for determining the best route a mobile robot can follow in any environment from its origin to its target destination. The objective of a kinematic controller is to follow a trajectory described by its position and velocity profiles as function of time. Many researchers have studied kinematic behaviour and provided some adequate solutions for (kinematic) motion control of a mobile robot system. Most of controllers of mobile robot are not considering the dynamics of the system.

Following the example of our intelligence, the robot explores the environment and identifies human understandable guiding clues to find a way to the assigned destination. The aim of this research is to idealize an existing autonomous mobile robot, on levels like kinematics, perception, cognition, sensor fusion, path analysis, path planning and navigation. This thesis is enthused to the goal of design and development of Autonomous mobile robot enriched with a distinctive control skill such that robot has the ability:

- ❖ To move freely in its environment,
- ❖ To perform a number of different tasks,
- ❖ To learn from experience and change its behaviour accordingly,
- ❖ To build internal representation of its world that can be used for reasoning processes like navigation,
- ❖ Finally, to choose the most appropriate suggestions adequate to human intelligence for finding a way to the consigned endpoint.

As a trial to move the robot through the given environment, fuzzy control concept has already proven to be worthwhile in both global and local path planning tasks for autonomous mobile objects. A set of linguistic fuzzy rules are developed to implement expert knowledge under various situations. Sensor signals are fed to the controller and the output provides motor control commands (e.g. turn left or right). Under the control of the proposed fuzzy logic-based model, the mobile robot can generate paths towards the target by integrating different preliminary robotic behaviours. The artificial life approach to evolutionary robotics is specifically designed to grow a neural structure with complex dynamical properties for path recognition of autonomous mobile robot. Neural networks are often used to enhance and optimize the outcome of fuzzy logic based system, e.g. by introducing a learning ability. This learning ability is achieved by presenting a training set of different examples to the network and using learning algorithm, which changes the weights (or the parameters of activation functions) in such a way that network will reproduce a correct output for the input values associated with nonlinearities.

In order to overcome the limitations of fuzzy logic introduced to mobile robots, type 2 fuzzy logic is introduced which enables the robot to gain a better control over the environment and reach the goal more efficiently and effectively in a shorter time using the shortest path. The path obtained is compared to the path obtained from other mechanisms to reveal the best of all. Even though human being is considered to be the intelligent of all animals, some animals have better organisation and path seeking skills than the humans. An example of this can be seen in case of the honey bees which move in search of food and gather their food in a single place thereby allowing others to take up their path and follow them to their food position. This algorithm when applied to a mobile robot enables it to move freely in the environment but search for a more complacent path to reach the destination.

1.3 THESIS STRUCTURE

The practices as organized in this thesis are approximately divided into ten chapters.

- Succeeding the introduction, Chapter 2 puts on the literature review of foregoing investigations on kinematics and analysis of mobile robot configuration, fuzzy logic controller, neural network backpropagation algorithm and a newly designed artificial bee colony algorithm.

- Chapter 3 studies the kinematics architecture of mobile robot configuration for weighing performance of the model robot pertaining to different mechanical aspects.

- Chapter 4 studies the concept of fuzzy logic to design a reactive behavioural controller whose performance has also been assessed.

- Chapter 5 discusses the backpropagation algorithm and neural networks in order to build a controller that would allow better navigation of the robot using training patterns.

- Chapter 6 aims at the study of type 2 fuzzy logic, an enhancement of fuzzy logic behaviour, that will allow the formation of a better and certain path by getting rid of excess uncertainties.

- Chapter 7 gives a detailed study of the artificial bee colony algorithm, an evolutionary algorithm, that enables the mobile robot to reach the destination using a more suitable path by studying the behaviour of honeybees.

- Chapter 8 describes hardware aspect of a simple mobile robot configuration by accumulating different sub modules.

- Chapter 9 discusses a comprehensive description of results and discussion has been carried out.

- Chapter 10 gives the contributions and conclusions of this research and future directions for further investigation has also been conferred.

The paper published related to the thesis has been listed at the last.

CHAPTER 2

LITERATURE REVIEW

2. LITERATURE REVIEW

Crafting a robust global navigation technique for economical mobile robot has been a big challenge for scientists for many years. There is a great number of potential applications for autonomous mobile robots in indoor environments, extended from cleaning, to surveillance, to search and rescue operations in burning buildings or hostage situations, to assisting the handicapped or elderly around the home. In order to realize these applications, all difficulties and challenges in this domain must be focused. The progress made in the past decades in the field of kinematics and dynamic modelling and design of artificial intelligence techniques used for navigation of mobile robots are briefly reviewed here.

2.1 INTRODUCTION

Autonomous mobile robots have the ability to move in its environment and performing a number of different tasks by adapting to the changes in environments and learning from experience in order to change behaviour accordingly and last but not the least to build an internal representation of its world that can be used for reasoning process like navigation. Amongst several issues related to autonomous operation, previous research works on two main computational issues are elaborated here: Modelling of Mobile Robot and Path planning and Navigation of the mobile robot. Modelling of mobile robots requires previous analysis of the kinematic and dynamic constraints whereas navigation can be considered as a process whose inputs are the specifications taken from the environment, description of the current and future position, description of the destination and the agent's observations of the environment. The produced output is the most appropriate movement orders to reach the destination position, avoiding obstacles and other exception situations that can arise.

Much research has been done on many aspects related to mobile robots. A literature review cannot simply be a catalogue of all the articles published on a subject, the list would be much too long and could not be include each contribution. The alternative is to include in this chapter only those contributions that cover to kinematics stability of mobile robots which provides desired trajectory and artificial intelligence technique that helps to design an intelligent controller for robot. A large number of researchers have used kinematic models to develop motion control strategy for mobile robots. The ultimate goal of mobile robotics research is to provide the robots with high intellectual ability, by which navigation in an unknown environment is achieved using on line sensory information. It summarizes the past work, mostly in computational geometry and robotics, and discusses possible directions for research. This chapter provides details survey report within important aspects of research work to seek out optimal path and track the target in the competing clutter environment on the basis of sensory data and their structural significance using fuzzy logic, backpropagation algorithm, type 2 fuzzy logic and artificial bee colony algorithm.

Another challenge in literature review is that even the perception of what constitutes progress varies widely in the research community. The representations would be difficult to extend other scenarios where a robot may need to seek out optimal path and track the target in the competing clutter environment on the basis of their semantic significance. Despite these challenges, the next sections review in this article and highlights some of the more interesting, important and experimental milestones. This chapter provides details survey report within important aspects of what the researchers have worked in the area of navigational path analysis and planning for mobile robot using fuzzy logic, neural network, adaptive neuro-fuzzy and heuristic rule base neural network technique.

2.2 KINEMATIC ANALYSIS OF WHEELED MOBILE ROBOT

The kinematic model of a mobile robot is mainly the description of the admissible instantaneous motions in respect of the constraints. On the other hand, the dynamic model accounts for the reaction forces and describes the relationship between the above motions and the generalized forces acting on the robot. These models can be expressed in a canonical form which is convenient for design of planning and control techniques. This section provides a detailed survey report of kinematics of mobile robot. With reference to the unicycle kinematics, this part reviews several control strategies for trajectory tracking and posture stabilization in an environment free of obstacles. A kinematic methodology is the first step towards achieving these goals.

Mobile robots are more efficient than legged or treaded robots on hard as well as smooth surfaces, and have potential enough to find widespread application in industry, because of the hard, smooth plant floors in existing industrial environments [1]. Several configurations for mobility can be found in the applications as mentioned by Jones et al. [2]. The most common form single-body robots are the differential drive and synchronic drive tricycle or car-like drive, and omnidirectional steering robots [3]. Besides the relevance in applications, the problem of autonomous motion planning and control of mobile robot has attracted the interest of many researchers to view its theoretical challenges [4]. The motion control of wheeled mobile robots has been able to draw considerable attention over the past few years. The nonholonomic behaviour in robotic systems is particularly interesting; since it points out that the mechanism can be completely controlled by using a reduced number of actuators. Particularly, these systems are typical examples of nonholonomic mechanisms due to the perfect application of the rolling constraints on the wheel motion [5]. Several controllers have been proposed for the motion control of mobile robots with nonholonomic constraints, where the two main approaches to controlling mobile robots are posture stabilization and trajectory tracking.

The procedure of modelling can be inspired by definition of a wheeled mobile robot according to Muri and Neuman [6] as this, "A robot capable of locomotion on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface. A wheel assembly is a device that provides a relative motion between its mount and a surface on which it is intended to have a single point of contact." However it is required that the vehicle kinematic design has the appropriate degrees of freedom (mobility) so that it can adapt to the variations in the surface and the wheels roll without slip. Mobility is enhanced by the use of omnidirectional wheels instead of conventional wheels [7]. The necessity of ideal rolling without sideways slipping for wheels enforces non-holonomic (non-integrable) constraints on the motion of the wheels of mobile robot [8]. The relation between the rigid body motion of the robot and the steering and drive rates of wheels has been developed by Alexander and Maddocks [9] based on constraint as "rolling without sliding". Slippage due to misalignment of the wheels is investigated here by minimization of a non-smooth convex dissipation functional that is derived from Coulomb's Law of friction. This minimization principle is equivalent to the construction of quasi-static motions.

Three related but different kinematical aspects have to be considered when designing a robot. They can be listed as mobility, control and positioning [10, 11]. The first one, mobility, deals with the possible motions that the robot can follow in order to reach its final destination in any orientation. The second aspect, control, relates to the choice of the kinematical variables: generalized velocities or coordinates. Finally, the third aspect, positioning, considers the localization system that is used to estimate the actual position and orientation of the robot by reducing the robot's region of uncertainty based on sensor measurements necessary to achieve an autonomous operation [12].

The motion along the configuration space is limited using the kinematic constraints. Kinematic limitations can be applied at any speed, while dynamic constraints are important to apply as an agent operates at higher speeds. Robot design has to tackle agent dynamics issues, as even a holonomic robot without any kinematic constraints will have to face some form of dynamics limitations, and in particular bounds on acceleration and velocity. Dynamics constraints limit the acceptable values for derivatives of an agent's position over time

Moon et al. [13] have proved that a wheeled mobile robot is not able to move along a straight line exactly, even if the kinematic problems are corrected perfectly, and this phenomenon is related to acceleration constraints on motor controllers. Kinematic model of a parallel wheeled mobile robot fails to meet Brockett's necessary condition for feedback stabilization thereby implying that no smooth or continuous time invariant. Stabilization and control of nonholonomic systems with dynamic equations have been considered in [14] whereas back stepping based methods are presented in several papers [15, 16, 17].

Internal error occurs from unsuitable setting up of the parameters and the time constant. External error inescapably appears when a WMR is being driven and it occurs by virtue of the two driving wheel's different friction and radius. In order to minimize such errors, Chung et al. [18] has proposed a feedback controller having two separated feedback loops; one of which is a position feedback, and the other an orientation feedback.

Based on back stepping algorithm, a robust adaptive controller has been proposed in [19, 20] to design an auxiliary wheel velocity controller in order to make the tracking error as small as possible as compared to the uncertainties in the kinematics of the robot and fuzzy logic techniques employed to learn the behaviours of the unknown dynamics of the robot and the wheel actuators. The major advantage of this method is that previous knowledge of the robot kinematics and the dynamics of the robot and wheel actuators is unnecessary. The parameters characterizing the robot dynamics are to be updated online, thereby providing smaller errors and better performance in applications in which these parameters can vary, such as load transportation. The stability of the whole system is analyzed using Lyapunov theory, and the control errors are ultimately bounded [21].

Deng et al. [22] designed a combined feedback control scheme based on Lyapunov function candidate [23] has been discussed for four obstacle cases in dynamic environments considering local minima problem. The controller includes virtual attractive force, repulsive force and detouring force, whereas the potential field function used for the design of the controller considers the Euclidean distance information and the magnitude information of the relative velocity between the robot and the target [24].

A dynamic model of a two-wheeled mobile robot has been derived in [25, 26] which shows that the translational motion and the rotational motion with 3 degrees of freedom of the body and here, the dynamic model is reduced to the kinematic model under certain assumptions. Arvin et al. [27] have presented mobile robots motion control technique based on pulse-width modulation (PWM).

The wheels of mobile robot have been modelled as a torus by Chakraborty and Ghosal [28] and used as a passive joint thereby enforcing a lateral degree of freedom so as to get a slip free motion in an uneven terrain without using variable length axle (VLA) as it has several limitations in application. Zhang et al. [29] have developed a feedback control law [30, 31], allowing a 2-wheel differentially driven mobile robot to track a prescribed trajectory by using the integral backstepping method and Lyapunov function for ensuring a trajectory tracking controller with global asymptotic stability.

Zohar et al. [32] recently proposes control schemes for trajectory tracking of mobile robot model which includes kinematic and dynamic effects on motion by using the notion of virtual vehicle [33] and the concept of flatness [34], and applying the backstepping [35] methodology.

Gandhi and Ghorbel [36] have proposed the harmonic drive system for non-linear controller to compensate for kinematic error in the presence of flexibility in high-speed regulation and trajectory tracking application. Pathak et al. [37] have discussed the behaviour of space robots with torque and attitude controller. A single curvature trajectory, having a constant and large rotation radius, has been proposed by Han et al. [38] as an optimal trajectory, in order to minimize the tracking error of the differential drive mobile robot while capturing a moving object along with the pre-determined initial and final states. A receding horizon controller may be used for tracking control of wheeled mobile robots subject to nonholonomic constraint in the environments without obstacles. The control policy is derived from the optimization of a quadratic cost function, which penalizes the tracking error and control variables in each sampling time [39, 40].

2.3 NAVIGATION USING FUZZY LOGIC CONTROLLER FOR MOBILE ROBOT

Fuzzy Logic technique plays an important role in the designing of an intelligent controller for mobile robot. This technique is used for navigation of mobile robots. Fuzzy set theory provides a mathematical framework for representing and treating uncertainty in the sense of vagueness, imprecision, lack of information and partial truth. Fuzzy control systems employ a mode of approximate reasoning that resembles the decision-making process of humans. A fuzzy system is usually designed by interviewing an expert and formulating the implicit knowledge of the underlying process into a set of linguistic variables and fuzzy rules. In particular for complex control tasks, obtaining the fuzzy knowledge base from an expert is often based on a tedious and unreliable trial and error approach [41]. Fuzzy set theory was introduced by Lofti Zadeh in the mid sixties. In 1965 Lotfi Zadeh proposed fuzzy set theory, and published a paper [42]. Fuzzy logic has been applied to diverse fields, from control theory to artificial intelligence. This section presents a variety of fuzzy logic techniques which address the challenges posed by autonomous robot navigation.

Autonomous mobile robot navigation in uncertain and dynamic environments demands adaptation and perception capabilities. Reactive control strategies imply a strong dependency on sensed information about the robot's environment. Thus, imprecision and uncertainties in perception from sensors have to be considered [43]. While the rules are based on qualitative knowledge, the membership functions defining the linguistic terms provide a smooth interface to the numerical process variables and the set-points [44]. Stability analysis of fuzzy systems is a very important research field in fuzzy systems practically from the pioneer work of Mamdani on fuzzy control applications [45]. A Mamdani controller is usually used as a feedback controller. Since the rule base represents a static mapping between the antecedent and the consequent variables, external dynamic filters must be used to obtain the desired dynamic behaviour of the controller [46]. The control protocol is stored in the form of if-then rules in a rule base which is a part of the knowledge base. While the rules are based on qualitative knowledge, the membership functions defining the linguistic terms provide a smooth interface to the numerical process variables and the set-points [47].

Intelligent control plays an important role when employing mobile robots in unstructured, unknown, and dynamic environments. The task complexity of intelligent control is greatly reduced by dividing the overall task into subtasks. These subtasks are modelled as perception-action units, called behaviours. The reduced task complexity in a behaviour-based approach increases responsiveness to environmental dynamics [48]. Systems equipped with fuzzy logic controllers give rise to nonlinear dynamic systems. This theory provides an overall perspective on the behaviour modes of the system, which can be used as a guide for the search of concrete behaviours [49].

Fuzzy systems belong to the family of nonlinear systems and they can have, in general, a complex analytical description [50]. It is not easy and time consuming for human experts to examine all the input-output data from a mobile robot to find a number of proper rules for a fuzzy controller. To copy with this difficulty, an intelligent mobile robot with automatic fuzzy controller design approaches is necessary [51]. It should also be noticed that although the operating range of the input is restricted by the saturation, the range of the other system variables cannot be bounded. This is in fact the cause of the troubles with the nonlinear nature of the saturation [52]. In this context, fuzzy logic is often adopted to overcome the difficulties of modeling the unstructured, dynamically changing environment, which is difficult to express using mathematical equation [53]. A class of fuzzy control laws can be formulated using the Lyapunov's direct method, which can guarantee the convergence of the steering errors [54].

The fuzzy controller can be optimised by using the schema co-evolutionary algorithm, which finds an optimal solution [55]. The main problem in fuzzy control involves the design of the fuzzy knowledge base. Various approaches to this problem have been proposed, including trial and error. For a mobile robot to intermesh navigation in various environments using fuzzy logic controller represents significant progress for the entire research community. An adaptive-resonance theory based fuzzy controller, including an adaptive-resonance theory based environment recogniser, a comparer, combined rule bases, and a fuzzy inferring mechanism, is introduced for the purpose of the adaptive navigation of the quadruped robot[56].

The adaptive fuzzy logic control based on physical properties of wheeled inverted pendulums makes use of a fuzzy logic engine and a systematic online adaptation mechanism to approximate the unknown dynamics [57]. Fuzzy adaptive extended information filtering is to improve estimation accuracy and robustness for the localization system, while the system lacks sufficient information of complete models or the process and measurement noise varies with time [58]. The unmanned control of the steering wheel is, at present, one of the most important challenges facing researchers in autonomous vehicles within the field of intelligent transportation systems [59]. Once this control architecture has been implemented, installed, and tuned, the resulting steering maneuvering is very similar to human driving, and the trajectory errors from the reference route are reduced to a minimum. In the controller a rule base of positive rules can be specified by an expert for directing the vehicle to the target in the absence of obstacles, while a rule base of negative rules can be experimentally determined from expert operation of the vehicle in the presence of obstacles [60].

Fuzzy logic system promises an efficient way for obstacle avoidance. However, it is difficult to maintain the correctness, consistency, and completeness of a fuzzy rule base constructed and tuned by a human expert. Reinforcement learning method is capable of learning the fuzzy rules automatically [61]. Martinez et al. [62] have considered a problem which is consisted of achieving sensor based motion control of mobile robot among obstacles in structured and unstructured environments with collision-free motion. Sensor-based navigation method, which utilised fuzzy logic and reinforcement learning for navigation of mobile robot in uncertain environments, has been proposed by Boem et al. [63] they have discussed about the navigation of mobile robot using fuzzy logic.

The concepts of car maneuvers, fuzzy logic control, and sensor-based behaviours are merged to implement the human-like driving skills by an autonomous car-like mobile robot.

Four kinds of Fuzzy logic controller, fuzzy wall-following control, fuzzy corner control, fuzzy garage-parking control, and fuzzy parallel-parking control, are synthesized to accomplish the autonomous fuzzy behaviour control [64]. The architecture for the fuzzy controller is a hierarchical scheme which combines seven modules working in series and in parallel [65]. The scaling factors and the coefficients of the sliding surface for the control of the steering angle and forward–backward velocity of a car-like mobile robot are adopted by that for the control of two motors [66]. Wang [67] has used fuzzy systems to model higher levels of hierarchical systems and design controllers for the hierarchical systems. Seraji's [68] paper presents a new strategy for behaviour- based navigation of field mobile robots on challenging terrain. Outdoor environments are particularly challenging for mobile robots as they offer dynamic, unstructured, and highly variable situations where the inconsistency of the terrain, the irregularity of the product, and the open nature of the working environment result in complex problems of identification, modeling, sensing, and control [69].

One important problem in autonomous robot navigation is the effective following of a unknown path traced in the environment in compliance with the kinematic limits of the vehicle, i.e., bounded linear and angular velocities and accelerations. In this case, the motion planning must be implemented in real time and must be robust with respect to the geometric characteristics of the unknown path, namely curvature and sharpness [70]. The stabilizing controller is designed as a state optimal controller and second application is the optimization method applied to the design of a fuzzy controller for vision-based mobile robot navigation [71].

The fuzzy error correction control system can be used to navigate a robot along an easily modifiable path in a well-structured environment. The fuzzy engine gives outputs commands for the robot wheels. These commands determine the necessary angle of rotation to correct the direction of travel in order for the robot to remain on the path [72]. Das et al. [73] have assumed a control structure that makes possible the integration of a kinematic controller and an adaptive fuzzy controller for trajectory tracking for nonholonomic mobile robots. The hybrid controller is able to choose a better position according to the circumstances encountered [74].

The information about the global goal and the long-range sensory data are used by the first layer of the planner to produce an intermediate goal, referred to as the way-point that gives a favourable direction in terms of seeking the goal within the detected area. The second layer of the planner takes this way-point as a sub goal and, using short-range sensory data, guides the robot to reach the sub goal while avoiding collisions [75]. Designing the controller on account of nonholonomic constraints gain more accurate position and velocity control, a self-organized fuzzy controller can be used to find solutions of optimal fuzzy input and output membership functions [76] and to determine a rule base process.

The fuzzy multi sensor data fusion scheme provides a novel mechanism to efficiently integrate task scheduling, action planning and motion control in a unified framework. The theoretical development of a complete navigation problem of an autonomous mobile robot is the situation for which the vehicle tries to reach the endpoint is treated using a fuzzy logic controller [77]. An efficient design methodology that allows starting with any kind of fuzzy controller and subsequently transforming it until a system suitable for easy digital signal processing implementation is obtained [78]. Navigation based on processing some analog features of Radio Frequency Identification signal is a promising alternative to different types of navigation methods in the state of the art. The main idea is to exploit the ability of a mobile robot to navigate a priori unknown environments without a vision system and without building an approximate map of the robot workspace, as is the case in most other navigation algorithms [79].

In the soccer game strategy Radio Frequency data transmitter is used to communicate among robot [80]. The development of the controllers is carried out by means of a reconfigurable platform based on field-programmable gate arrays. This platform combines specific hardware to implement fuzzy inference modules with a general-purpose processor, thus allowing the realization of hybrid hardware/software solutions [81]. The merger method is applied to fuzzy rule base simplification by automatically replacing the fuzzy sets corresponding to a given cluster with that pertaining to cluster prototype [82]. Target tracking requires team coordination to maintain a desired formation and to keep team-mates and target together. Generally, distributed autonomous systems using multiple robots are considered superior to others in terms of reliability, expandability, and flexibility. In contrast to a single robot system; they provide increased robustness by taking advantage of inherent parallelism and redundancy. Moreover, the versatility of a multi-robot system can provide the heterogeneity of structures and functions required to undertake different missions in unknown environmental conditions [83]. Research in autonomous multi-robot systems often focuses on mechanisms to enhance the efficiency of the group through some form of cooperation among the individual agents. One of the greatest challenges in robotics is to create machines that are able to interact with unpredictable environments in real time [84]. Intriguingly, a similar relationship between group size and efficiency has been documented in social robots.

2.4 NAVIGATION USING BACKPROPAGATION ALGORITHM

The human brain is the structure of marvel for everyone as it is very complex, nonlinear and parallel computer. It works on the basis of neural network and has neurons as its basis [85]. There are billions and trillions of neurons and the connections are more than them. The neural network system turns more interesting because of the fact that it has been adopted from the living beings [86]. There are billions of neurons and trillions of connections between them. The interest in neural network increases by the wish of understanding principles leading to the comprehension of the basic human brain functions, and to building the machines that are able to perform complex tasks [87]. Neural network theory revolves around the idea that certain key properties of biological neurons can be extracted and applied to simulations, thus creating a simulated brain [88]. There is a significant interest in autonomous mobile robots which may be defined as vehicles that are capable of intelligent autonomous navigation.

The key factor is that the robots must be able to understand the structure of the environment [89]. In order to reach their targets without collisions, the robots must be endowed with perception, data processing, recognition, learning, reasoning, interpreting, and decision-making and action capacities. Interest in neural networks emerged after the introduction of simplified neurons by McCulloch and Pitts [90]. In 1949 Hebb [91] formed the basis of 'Hebbian learning', now regarded as an important part of neural networks theory [92]. Rosenblatt [93] constructed neuron models in hardware during 1957. These models ultimately resulted in the concept of the Perceptron. It has been an important development and the underlying concept is still in use today widely. Widrow and Hoff [94] presented simplified artificial neuron development. Most neural network researchers left the field when Minsky and Papert published their book Perceptrons in 1969 [95] in which they showed the deficiencies of perceptron models,

Neural networks are parts of intelligent controllers and also parts of well-known structures [96]. They are adaptive and statistical structures that are based on an analogy with the brain structure they are referred to as adaptive because they can learn to estimate the parameters of some population using some number of examples at a time. They do not differ from standard statistical models [97] and hence can be used as statistical tools in a various number of fields like statistics, engineering, econometrics etc.

If human can understand animal behaviour control, and comparable technology is available, then it would be possible to build a robot that behaves the same way. Recent advances in both knowledge and technology have begun to make this possibility a realistic aim in invertebrate neuroscience [98, 99, 100]. Neural network circuits can produce coordinated patterns of high-dimensional rhythmic output signals while receiving only simple, low dimensional, input signals [98]. A growing number of studies have been done in which hypotheses for the behavioural function of neural circuits have been tested by implementing them as controllers for robots and evaluating the robot behaviour [101]. An artificial neural network is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks [102]. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation [103].

The neural networks can be used to model complex relationships between inputs and outputs or to find patterns in data [104]. However, the evolved neural controllers could be flimsy in inexperienced environments, especially in real worlds, because the evolutionary optimization processes would be executed in idealized simulators. This is well known as the gap problem between the simulated and real worlds. To overcome this, Kondo [105] focused on an evolving on-line learning ability instead of weight parameters in a simulated environment.

Basically, the control of a robot arm and the control of a mobile robot are similar to any controller [106]. First a path is planned and the path is transformed from Cartesian domain to the joint or wheel domain using the inverse kinematics of the system and finally a dynamic controller maps the set points in this domain to actuator signals. However, in practice the problems with mobile robots occur more with path-planning and navigation than with the dynamics of the system. A new paradigm of cognitive science has emerged [107] whose hallmark is to focus on the situated and embodied nature of intelligence. Research in so-called behaviour-based artificial intelligence [108], embodied neurobiology, and embodied cognitive science [109] has challenged the traditional view according to which intelligence is an abstract, symbolic process independent of physical implementation.

The artificial life approach to evolutionary robotics is used as a fundamental framework for the development of a modular neural control of autonomous mobile robots

[110]. The applied evolutionary technique is especially designed to grow different neural structures with complex dynamic properties which is due to a modular neuro dynamics approach to cognitive systems, stating that cognitive processes are the result of interacting dynamical neuro-modules [111]. Relevant brain centres, known as Mushroom Bodies and Central Complex have been recently identified in insects: though their functional details are not yet fully identified, it is known that they provide secondary pathways allowing the emergence of cognitive behaviours [112]. In recent years, mobile robots have been required to become more and more autonomous in such a way that they are able to sense and recognise the three dimensional space in which they live or work [113]. Werbos et al. [99] reviewed the empirical results which fit the theory, and suggested important new directions for research, within the scope of NSF's recent initiative on cognitive optimization and prediction.

Basically, neural networks are built from simple units, called neurons or cells by comparing with the real thing. These units are connected by a set of weighted connections [114]. Learning is generally accomplished by modification of the connection weights. Each unit corresponds to a feature or a characteristic of a pattern that we want to analyse or that we want to use as a predicator.

These networks usually organize their units into several layers. The first layer is called the input layer, the last one the output layer. The intermediate layers are called the hidden layers. The information to be analysed is fed to the neurons of the first layer and then propagated to the neurons of the second layer for further processing [115]. The result of this processing is then propagated to the next layer and then to the next layer until the last layer. Each unit receives some information from other units and processes this information, which will be converted into the output of the unit.

The goal of the network is to learn and discover an association between input and output patterns and to analyse or find the structure of the input patterns. The learning process is achieved through the modification of the connection weights between units. In statistical terms, this is equivalent to interpreting the value of the connections between units as parameters (e.g., like the values of a and b in the regression equation by = a + bx) to be estimated [116]. The learning process specifies the "algorithm" used to estimate the parameters.

Rumelhart and McClelland [117] formulated the backpropagation algorithm and used it in layered feed forward neural networks. As the name itself suggests, this means that the artificial neurons are organized in layers, and send their signals "forward", and then the errors are propagated backwards. The network receives inputs by neurons in the *input layer*, and the output of the network is given by the neurons on an *output layer*. There may be one or more intermediate *hidden layers*.

The purpose of the learning rule is to train the network to perform some task. There are many types of neural network learning rules [181]. They fall into three broad categories: supervised learning, unsupervised learning and reinforcement learning. The mobile robot navigation deals with application of back propagation algorithm in both, supervised and reinforcement learning approaches [119]. A hybrid approach for the autonomous motion control of robots in cluttered environments with unknown obstacles is introduced by Maravall et al. [120].

Decision making system is the most important part of the robot soccer system [121]. As the environment is dynamic and complex, one of the reinforcement learning methods is employed in learning the decision-making strategy. Nelson et al. [122] have described the evolutionary training of artificial neural network controllers for competitive team a game playing behaviours by teams of real mobile robots. A neural network based machine vision system, which is intended to act as a reconfigurable inspection tool and used in manufacturing environments [123, 124]. Discriminative training is accomplished in a supervised manner, using gradient-descent method. The approach is suitable for navigation and for map learning [125]. Many current machine learning paradigms has been used for this purpose, but, result in opaque models are difficult, if not impossible to analyse, which is an inhibition in safety-critical applications or application scenarios where humans and robots occupy the same workspace [126].

The hybrid architecture using band pass filtering, cross-correlation and recurrent neural networks can be used to develop a robust, accurate and fast sound-source localisation model for a mobile robot [127]. The new approach in robotic learning systems was proposed by Burgsteiner et al. [128]. It provided a method to use a real-world device that operates in real time, controlled through a simulated recurrent spiking neural network for robotic experiments. Robot path-planning techniques can be divided into two categories. The first, called local planning relies on information available from the current 'viewpoint' of the robot. This planning is important, since it is able to deal with fast changes in the environment.

The second situation is called global path-planning, in which the system uses global knowledge from a topographic map previously stored in memory. Even though global planning permits optimal paths to be generated [129], it has its weakness. The third generation of artificial neural networks, spiking neural networks [130], have unique advantages and are good candidates for robot controllers. In the controller, the integrated-and firing model can be used and the Spiking neural network is trained by the Hebbian learning algorithm [131]. The transportation using wheels is one of the most popular transportation mechanisms for mobile robots because of its high energy efficiency, simple mechanisms and well-investigated control systems [132]. Wheel type mobile systems are the most popular transportation mechanisms because the energy efficiency is high while the mechanism is simple and the control system is well investigated [133]. On the other hand, the wheel type mobile robots have difficulties in rough terrain movement. Perception and behaviour are usually considered to be separate processes. Behavioural learning forms associations between perception and action, organized by reinforcement, without regard for the construction of perception [134]. The behaviour is organized as a dynamic hierarchy of independent schemas [135].

The backpropagation algorithm uses supervised learning method [136]. This means that an algorithm is provided with examples of the inputs and outputs that the network is supposed to compute, and then the error (difference between actual and expected results) is calculated. The idea of the backpropagation algorithm is to reduce this error, until the ANN *learns* the training data [117]. The training begins with random weights, and the goal is to adjust them so that the error will be minimal. The activation function of the artificial neurons in ANNs implementing the backpropagation algorithm is a weighted sum [137]. If the output function would be the identity (output=activation), then the neuron would be called linear. But these have severe limitations. The sigmoidal function is very close to one for large positive numbers, 0.5 at zero, and very close to zero for large negative numbers. This allows a smooth transition between the low and high output of the neuron (close to zero or close to one). We can see that the output depends only in the activation, which in turn depends on the values of the inputs and their respective weights [138].

Now, the goal of the training process is to obtain a desired output when certain inputs are given [139]. Since the error is the difference between the actual and the desired output, the error depends on the weights, and we need to adjust the weights in order to minimize the error. The backpropagation algorithm now calculates how the error depends on the output, inputs, and weights [140].

2.5 NAVIGATION USING TYPE 2 FUZZY LOGIC

The concept of a type-2 fuzzy set was introduced by Zadeh [141] as an extension of the concept of type-1 fuzzy set. Such sets are fuzzy sets whose membership values are themselves type-1 fuzzy sets. These are very useful in circumstances where there is difficulty in determining an exact membership function for a fuzzy set; hence, they are useful for incorporating linguistic uncertainties [142]. A fuzzy relation of higher type (e.g., type-2) has been regarded as one way to increase the fuzziness of a relation, and, according to Hisdal, "increased fuzziness in a description means increased ability to handle inexact information in a logically correct manner [142]". According to John, "Type-2 fuzzy sets allow for linguistic grades of membership, thus assisting in knowledge representation, and they also offer improvement on inferencing with type-1 sets [144]".

Type-2 sets can be used to convey the uncertainties in membership functions of type-1 sets, due to the dependence of the membership functions on available linguistic and numerical information. Linguistic information (e.g., rules from experts), in general, does not give any information about the shapes of the membership functions. When membership functions are determined or tuned based on numerical data, the uncertainty in the numerical data, e.g., noise, translates into uncertainty in the membership functions. In all such cases, information about the linguistic=numerical uncertainty can be incorporated in the type-2 framework. In [143], Liang and Mendel demonstrated (using real data) that a type-2 fuzzy set, a Gaussian with fixed mean and uncertain standard deviation (std), is more appropriate to model the frame sizes of I=P=B frames in MPEG VBR video tra<c than is a type-1 Gaussian membership function. When the secondary MFs are interval sets, we call them "interval type-2 fuzzy sets". The operations of interval type-2 fuzzy sets are studied in [145, 146].

Fuzzy sets have been around for nearly 40 years and have found many applications. However they suffer from certain problems [147]. These fuzzy sets are, in fact, type-1 fuzzy sets. Type-2 fuzzy sets are 'fuzzy fuzzy' sets and are more expressive [148]. Type-2 fuzzy sets and systems generalize (type-1) fuzzy sets and systems so that more uncertainty can be handled. From the very beginning of fuzzy sets, criticism was made about the fact that the membership function of a type-1 fuzzy set has no uncertainty associated with it, something that seems to contradict the word fuzzy, since that word has the connotation of lots of uncertainty.

A type-2 fuzzy set lets us incorporate uncertainty about the membership function into fuzzy set theory, and is a way to address the above criticism of type-1 fuzzy sets head-on. And, if there is no uncertainty, then a type-2 fuzzy set reduces to a type-1 fuzzy set, which is analogous to probability reducing to determinism when unpredictability vanishes. In order to symbolically distinguish between a type-1 fuzzy set and a type-2 fuzzy set, a tilde symbol is put over the symbol for the fuzzy set; so, A denotes a type-1 fuzzy set, whereas Ã denotes the comparable type-2 fuzzy set.

The membership function of a general type-2 fuzzy set, Ã, is three-dimensional where the third dimension is the value of the membership function at each point on its two-dimensional domain that is called its footprint of uncertainty (FOU).Unlike a type 1 fuzzy set, whose membership for each value is a number, the membership of a type 2 fuzzy set is an interval.

To go from an interval type-2 fuzzy set to a number two steps are required. The first step, called type-reduction, is where an interval type-2 fuzzy set is reduced to an interval-valued type-1 fuzzy set. There are as many type-reduction methods. An algorithm known as the KM Algorithm is used for type-reduction. Although this algorithm is iterative, it is very fast. The second step of Output Processing, which occurs after type-reduction, is called defuzzification. Because a type-reduced set of an interval type-2 fuzzy set is always a finite interval of numbers, the defuzzified value is just the average of the two end-points of this interval.

2.6 NAVIGATION USING ARTIFICIAL BEE COLONY

In past few decades, the computational researchers have been numerously interested to the natural sciences, and specifically biology, as source of modelling paradigms. Many research areas are massively influenced by the behaviour of various biological entities and phenomena [149]. It gave birth to most of population-based metaheuristic algorithms. They modelled the animal social behaviours such as ant, fish, bird, bee etc. They can be regarded as belonging to the category of intelligent optimization tools used to solve a computational and complex problem in different areas. Honey bees are one of the one of the deeply studied social insects. In the early years many studies based on the different bee behaviours have been developed to solve complex combinatorial and numerical optimization problems [150].

We can enumerate two sub-models in bee model. The first one, food source searching sub model, is based on the food source searching behaviour. Anew system called Bees system (BS) based on food source searching as foraging behaviour of bee colonies was developed. It was tested through many instances of the travelling salesman problem (TSP) [151].Another model was inspired by the nest site searching behaviour. The marriage behaviour represents this second model of the algorithms inspired by bee life. They rely on the same principle. It is the result of a synthesis study after a large bibliographic research on the various computational systems inspired by the different bee behaviours [152]. The first model represents the algorithms inspired by the foraging behaviour of bees. In its turn, it can be divided into two sub-models; the first one is based on the food source searching and the second represents studies which turn round the new nest site searching. The second model groups different algorithms which are inspired by marriage behaviour. There are other algorithms inspired by the evolution of the queen that can be considered as genetic algorithms improvement [153].

2.7 CONCLUSION

Firstly the kinematics and dynamic analysis of differential drive mobile robot has been addressed here, and the problem of model based constraints and trajectory tracking have been found in a number of research work. This chapter also provides a detailed review report which has been used in last decades by many researchers in the area of new intelligent control techniques like Fuzzy Logic and Fuzzy-Neural Network. Sensors used in different robotic application are also reviewed here. From the survey it has been perceived that the mobile robot navigation can be controlled successfully in a complex, unknown and dynamic environments using the above strategies.

CHAPTER 3

KINEMATIC ANALYSIS OF MOBILE ROBOT

3. KINEMATIC ANALYSIS OF MOBILE ROBOT

3.1 INTRODUCTION

Kinematics is the study of motion of points, objects or system of objects without taking into consideration the forces due to which the movement is caused. It is also referred to as geometry of motion. In other words, it is the study of behaviour of mechanical systems. Basically trajectories of points, lines and other geometric properties like velocity and acceleration are studied for the purpose of kinematic analysis. Kinematic analysis is used to measure the kinematics quantities used in describing the motion of an object. In case of mobile robots, kinematic analysis helps us understand the mechanical behaviour of robots in order to create control software for mobile robot hardware.

All mechanical systems require such an analysis to help them work better. Robotic manipulators have also been subjected to such an analysis for more than thirty years. Such manipulators were more complex than the ordinary robots and hence more deep analysis was required for their understanding. The questions about the kinematics of mobile robotics are the same as that posed for mobile manipulators.

The position estimation of the robot also has much dissimilarity in comparison to the position estimation of the robotic manipulator. In case of a manipulator, one end is fixed to the environment. Hence the position of the manipulator is taken as the position of the end effector. However, in case of a wheeled mobile robot the whole robot is movable and its position is not a factor to be measured instantaneously. Its position is to be estimated by integrating its motion over time and adding inaccuracies of motion estimation due to slippage constraint.

3.2 KINEMATIC CONSTRAINTS

The derivation of a model for the whole robot's motion is a bottom to top approach. The fact that each individual wheel contributes to the motion control of the robot as well as imposes constraints on the robot is predictable. However the wheels work together on the robot chassis geometry so that their constraints are also combined together to form the overall motion constraints of the robot chassis. However the constraints and forces of each wheel is to be expressed with respect to a lucid and reliable reference frame. The constraints imposed by the wheels are

- ❖ There should be a rolling motion and
- ❖ There should be no lateral slip

3.2.1 WHEEL DESIGN

The design of a wheeled mobile robot is concerned about the kinds of wheels and their configuration and actuation systems. The mobility characteristic of the robot is defined using these parameters.

The wheels can be divided into five major classes depending on their kinematics. Hence the wheels used in a robot's architecture affects the overall kinematics of the robot.

- ❖ Fixed standard wheel
- ❖ Steerable standard wheel
- ❖ Castor wheel
- ❖ Swedish wheel
- ❖ Spherical wheel

The wheel needs to be steered along a vertical axis if it needs to change its direction. So, the rotational axis of the wheel plays a vital role in the motion of the wheel.

The fixed standard wheel is fixed to robot chassis and moves only back and fro.

- ❖ There is no vertical axis of rotation or steering for the fixed standard wheel.
- ❖ The angle between the chassis and the wheel axis is fixed.

It is limited to move the robot back and forth along the wheel plane and rotation around its contact point with the ground plane. Such wheel is used in carts where the direction of the vehicle is changed by the movement of the bullocks.

The steerable standard wheel is fixed to the robot chassis but has a vertical axis of rotation. It is different from the fixed standard wheel in that it has an extra degree of freedom.

- ❖ There is a vertical axis of rotation.

- ❖ The angle between the chassis and the wheel axis is not fixed and changes with every steering.

Such wheels are used in all transportation vehicles as their direction is changed by the change in direction of the wheels.

The castor wheel is offset from the chassis. Hence its wheel of rotation does not pass through the ground contact point.

- ❖ There are two axes, one through the robot chassis and the other through the wheel.
- ❖ Both the axes have a gap between them.

Such wheels are used in shopping carts, office chairs and industrial material handling equipment. The Swedish wheel or the omni wheel has small discs around its circumference that are perpendicular to the rolling direction. Hence the wheel can roll with full speed as well as slide laterally easily.

- ❖ There is no vertical axis of rotation and the movement is done by adding an extra degree of freedom to the fixed standard wheel.

- ❖ The rollers attached to the wheel circumference have anti parallel axes to the main axis of the wheel.

Such wheels are used in holonomic drive systems and in robots which need to move in every direction. The spherical wheel has no direct axis of rotation and hence has no rolling or sliding constraints. It is omnidirectional.

- ❖ There is no principal axis of rotation.

- ❖ It can rotate in any direction.

Such wheels are used in computer mouse where powered rollers are placed against the top side of the sphere in order to provide rotational force.

3.3 WHEELED MOBILE ROBOTS

Wheels are the most suitable locomotion devices in case of robots and other man-made vehicles as they can achieve decent efficiency level and do not need a complicated mechanical implementation. Generally, wheeled mobile robots do not have an issue of instability in motion since a minimum of three wheels suffices the condition of stability. However, a robot with two wheels can also move steadily whereas in case of more than three wheels, a suspension system is required to maintain the robot's contact with the ground surface. Hence the main issue of concern is the traction provided by the wheel in order to cover every type of terrain that it passes through with the required velocity.

Depending on the orientation and types of wheels used, Wheeled Mobile Robots can be classified into

- Bicycle or tricycle – one independent orientable wheel and other independent fixed wheel
- Differential drive robot – one independent fixed wheel and other omnidirectional wheels
- Synchro drive robot – one independent orientable wheel and other omnidirectional wheels
- Omnidirectional robot – only Castor and Swedish wheels
- Two wheeled differential drive - Two independent orientable wheels

Depending on the number of wheels in the Wheeled Mobile Robot, Wheeled Mobile Robots can be divided into

- Two wheeled mobile robot
- Three wheeled mobile robot
- Four wheeled mobile robot
- Six wheeled mobile robot

3.4 POSITION OF THE ROBOT

The position of the robot has to be expressed in terms of a relation between the robot's global reference frame and the robot's local reference frame. Here X_I and Y_I define an arbitrary inertial basis on the plane as the global reference frame from the origin O $\{X_I, Y_I\}$. In order to specify the robot position we choose a point P on the robot chassis as the position reference point. $\{X_R, Y_R\}$ defines two axes relative to P on the chassis of the robot and is hence the robot's local reference frame. The position of the robot can be described using these coordinates x and y and the angular difference between global and local reference frames, given by θ. The position of the robot can be described using these factors described in the form of a matrix.

$$\xi_I = [\, x \quad y \quad \theta \,]^T \tag{3.1}$$

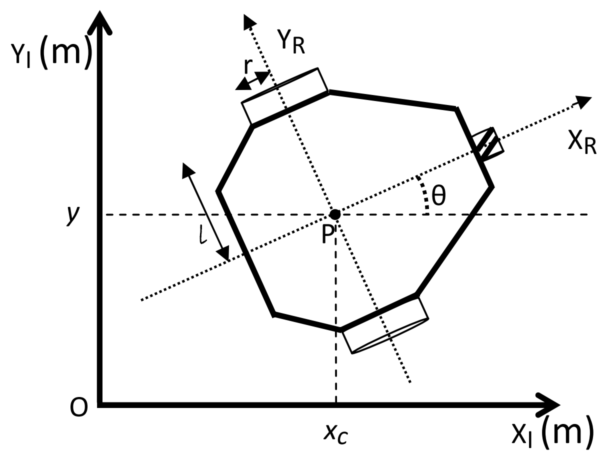


Fig. 3.1. Position of the robot in local reference frame

The matrix called the orthogonal matrix is used to map the robot's global reference frame in its local reference frame.

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

The operation of mapping global reference frame in terms of local reference frame is done using the orthogonal rotation matrix and is denoted by

$$R(\theta)\xi_I \tag{3.3}$$

3.5 FORWARD KINEMATIC ANALYSIS

A motion of a robot is a hierarchical process. In the given differential robot, two wheels each having diameter 'r' are present. A point P is located in the centre of two drive wheels, each at a distance l from P. If r, l, θ and speed of both the wheels, $\dot{\varphi}_1$ and $\dot{\varphi}_2$ respectively, are provided then a forward kinematic analysis of the robot will be able to predict the overall speed of the robot in the global reference frame.

$$\xi_I = \begin{bmatrix} x & y & \theta \end{bmatrix} = f\left(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2\right) \tag{3.4}$$

The contribution of each of the drive wheels in the local reference frame is first computed. The contribution of the spinning speed of each wheel to the translational speed at P in the direction of X_R is then calculated.

The approach leads us to first compute the contribution of each of the two wheels in the local reference ξ_R. Considering the contribution of each wheel's spinning speed to the translation speed at P in the direction of $+X_R$, if one wheel spins while the other wheel contributes nothing and is stationary then knowing that P is halfway between the two wheels, it will move instantaneously with half its speed. In a differential drive robot, these two contributions can be simply added to calculate the x_R component of ξ_R. As neither of the wheels contributes to sideways motion in the robot's reference frame, so y_R is always zero

Again, the contributions of each wheel are computed independently and just added for computing rotational component. Considering the right wheel (say wheel 1), a forward spin of this wheel results in counter clockwise rotation at point P. On the other hand, if wheel 1 spins alone, the robot pivots around wheel 2.

The rotation velocity ω_1 at P can be computed as the wheel is instantaneously moving along the arc of a circle of radius $2l$:

$$\omega_1 = \frac{r\dot{\varphi}_1}{2l} \tag{3.5}$$

The same calculation applies to the left wheel, with the exception that forward spin results in clockwise rotation at point P:

$$\omega_2 = \frac{-r\dot{\varphi}_2}{2l} \tag{3.6}$$

Combining these individual formulas yields a Forward Kinematic model for the differential-drive mobile robot in reference frame:

$$\dot{\xi}_R = \begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix} \tag{3.7}$$

Combining these individual formulas, a forward kinematic model for the differential-drive example robot can be framed:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix} \tag{3.8}$$

This approach to kinematic modelling provides information about the motion of a robot provided its component wheel speeds are given in straightforward cases. However, we wish to determine the space of possible motions for each robot chassis design. To do this, we must go further, describing formally the constraints on robot motion imposed by each wheel.

3.6 ANALYSIS OF WHEEL KINEMATICS

A differential drive robot has a maximum of three wheels to ensure stability of the mobile robot. Out of these three wheels, two separately controlled drive wheels are located at either side of the robot whereas one wheel is placed at the front or rear end of the robot in the center in order to stabilize the motion of the robot.

Both the drive wheels have a common horizontal axis which is fixed during the operation of the robot. The angular velocities of these wheels assure the mobility of the mobile robot. The third wheel is the Swedish wheel that is freely aligned and moves automatically on the route due to the forces developed by the two drive wheels. It assures the robot's equilibrium but does not contribute to the robot's movement.

The rotation of the vehicle about the robot's center of the axle is the result of the speed difference between two separately driven coaxial wheels. The coaxial wheels move in synchronous with one another to produce forward or reverse motion. This kind of robot is able to rotate in its position provided the angular velocities of the two coaxial wheels are equal and opposite. Here we assume that vertical motion is absent. As it is the mobile robot operates at relatively low speeds.

The analysis is simplified by further important assumptions. First assumption taken is that the wheel always remains in vertical position during the robot motion and there is no sliding at the only point of contact between the wheel and the ground. In other words the wheel is in rotation about the vertical axis through the contact point and the motion is under only pure rolling condition.

Using these assumptions, two constraints can be marked for each wheel type. The first constraint is to consider the concept of rolling contact that ensures the rolling of the wheel during motion in appropriate direction. The second constraint considers that there is no lateral slippage and no sliding orthogonal to the wheel plane.

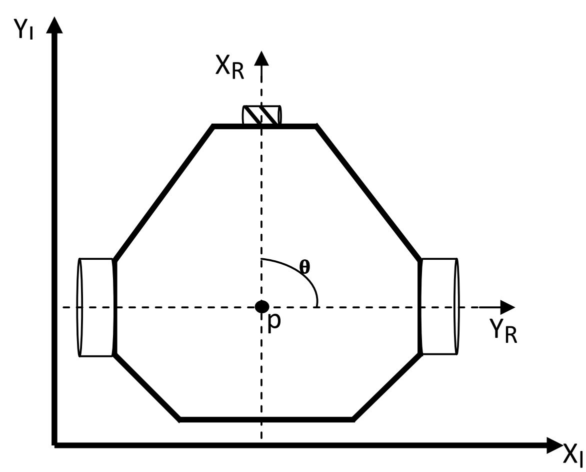


Fig. 3.2. Position of the robot in local reference frame

A fixed standard wheel and its position relative to the robot's local reference frame is seen in figure 3.3. The position of the robot is expressed in terms of polar coordinates by distance *l* and angle α while β denotes the angle of the wheel plane relative to the robot chassis. The angle β is fixed as the fixed standard wheel is non-steerable. If we consider a wheel of radius *r,* its rotational position around its horizontal axle is a function of time *t*: *φ (t).*

In order to get pure rolling at the point of contact using adequate amount of wheel spin the wheel imposes that all movement along the direction of wheel plane is given by:

$$\left[\sin(\alpha+\beta) - \cos(\alpha+\beta) - l\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \tag{3.9}$$

The sliding constraint for this wheel considers the wheel‘s motion normal to the wheel plane must be zero:

$$\left[\cos(\alpha+\beta) \;\; \sin(\alpha+\beta) \;\; l\sin\beta\right] R(\theta)\dot{\xi}_I = 0 \tag{3.10}$$

Now we have to compute the kinematic constraints of the robot chassis associated with the wheels. The principal idea is that each wheel imposes zero or more constraints on robot motion, and so the process is simply one of appropriately combining all of the kinematic constraints arising from all of the wheels based on the placement of those wheels on the robot chassis

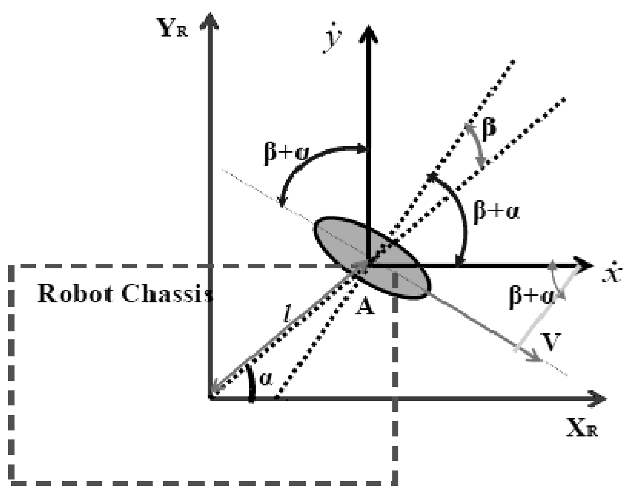


Fig 3.3. Fixed standard wheel and its parameters

As we know the fixed standard wheels have an effect on robot chassis kinematics and hence require to be considered while computing the robot‘s kinematic constraints. Supposing the robot has a total of N_f fixed standard wheels, β refers to the orientation of the N_f fixed standard wheels. As far as the wheel spin is concerned, the fixed wheels rotate around the horizontal axle and their rotational positions vary as a function of time, denoted as φ_f. The Swedish wheel on the rear end is unpowered and is free to move in any direction, so we ignore this third point of contact altogether as it does not impose any kinematic constraint.

Now the rolling constraints of all wheels can be combined into a unique expression and represented as:

$$J_{1f} R(\theta)\dot{\xi}_I - J_{2f}\dot{\varphi} = 0 \tag{3.11}$$

$$\Rightarrow \dot{\xi}_I = R(\theta)^{-1} J_{1f}{}^{-1} J_{2f}\dot{\varphi} \tag{3.12}$$

Here J_{1f} denotes a matrix of (N_f x 3) for all fixed standard wheels to their motions along their individual wheel planes, and J_{2f} is a constant diagonal matrix of (N_f X N_f) of all standard wheels radii.

In similar way we can also express the sliding constraints by combining all wheels into a single expression given by

$$C_{1f} R(\theta)\dot{\xi}_I = 0 \tag{3.13}$$

Here C_{1f} is of the order (N_f x 3).The above equation serves as a constraint for all standard wheels so that their components of motion orthogonal to their wheel planes must be zero. This sliding constraint over all fixed standard wheels has the most significant impact on defining the overall maneuverability of the robot chassis.

Combining (3.11) & (3.13) in a matrix form,

$$\begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_{2f} \\ 0 \end{bmatrix}\dot{\varphi} \tag{3.14}$$

In order apply the fixed standard wheel's rolling constraint formula, we have to first identify each wheel's α and ß values. Suppose that the robot's local reference frame is aligned in such a manner that the robot moves forward in the direction of $+X_R$.

In the present movement direction, for the right wheel α = -π/2 and β = π and for the left wheel α = π /2 and β=0. Note the value of β for the right wheel is necessary to ensure that positive spin causes motion in the $+X_R$ direction. Because the two fixed standard wheels are parallel, equation (3.10) results in only one independent equation. So, for the given values equation (3.16) can be written as

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \end{bmatrix} \\ \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_{1f} \\ 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} \tag{3.15}$$

$$\Rightarrow \dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} \tag{3.16}$$

Suppose that the robot is positioned such that $\boldsymbol{\theta} = \pi/3$, r =1 and l=1. If the robot engages its wheels unevenly, with speeds $\dot{\varphi}_1$ = 4 cm/s and $\dot{\varphi}_2$ = 2 cm/s, we can compute its velocity in the global reference frame:

$$\dot{\xi}_I = \begin{bmatrix} \cos\frac{\pi}{3} & \sin\frac{\pi}{3} & 0 \\ -\sin\frac{\pi}{3} & \cos\frac{\pi}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \end{bmatrix} \tag{3.17}$$

$$\Rightarrow \dot{\xi}_I = \begin{bmatrix} 0.75 \\ 1.29 \\ 0.5 \end{bmatrix} \tag{3.18}$$

3.7 MOBILE ROBOT MANEUVERABILITY

The overall manoeuvrability of a robot is a combination of the available mobility based on the kinematic sliding constraints of the standard wheels, in addition to the additional freedom contributed by steering and spinning of the steerable standard wheels.

3.7.1 DEGREE OF MOBILITY

The kinematic mobility of a robot chassis is its ability to move freely in the environment. The basic constraint limiting the mobility of the robot is the rule that every wheel must satisfy its sliding constraint. Therefore, we can formally derive robot mobility by starting from equation (3.13) which imposes the constraint that every fixed standard wheel must avoid any lateral slip.

Robot chassis kinematics is hence a function of the set of independent constraints arising from all standard wheels. The mathematical interpretation of independence is related to the rank of a matrix. Therefore rank of [C_{1f}] is the number of independent constraints. The greater the number of independent constraints, the greater is the rank of [C_{1f}], and hence more constrained is the mobility of the robot. We can define a robot's degree of mobility = δ_m where

$$\delta_m = 3\text{-rank}[C_{1f}] \tag{3.19}$$

The dimensionality of the null space (*dim N*) of [C_{1f}] matrix is a measure of the number of degrees of freedom of the robot chassis that can be immediately manipulated through changes in wheel velocity.

In the case of the differential drive robot in figure 3.1, the two wheels are aligned along the same horizontal axis. In fact, the second wheel imposes no additional kinematic constraints on robot motion since its zero motion line is identical to that of the first wheel. Differential-drive chassis has only one independent kinematic constraint. Therefore, rank [$C1f$] = 1 and δ_m=2. This fits with intuition: a differential drive robot can control both the rate of its change in orientation and its forward/reverse speed, simply by manipulating wheel velocities.

3.7.2 DEGREE OF STEERABILITY

The degree of mobility defined above quantifies the degrees of controllable freedom based on changes to wheel velocity. Steering can also have an eventual impact on a robot chassis pose, although the impact is indirect because after changing the angle of a steerable standard wheel, the robot must move for the change in steering angle to have impact on pose.

As with mobility, we care about the number of independently controllable steering parameters when defining the degree of steerability δ_s, but it deals only with steerable wheels. As we have taken the differential drive along with only the fixed standard wheels, so here δ_s = 0, i.e. the robot has no steerable standard wheels.

3.7.3 MANEUVERABILITY MEASUREMENT

The overall Degrees of Freedom (DOF) that a robot can manipulate is called the degree of maneuverability (δ_M). Thus the maneuverability comprises with the degrees of freedom that the robot changes its position directly through wheel velocity and the degrees of freedom that it indirectly manipulates by changing the steering configuration and moving.

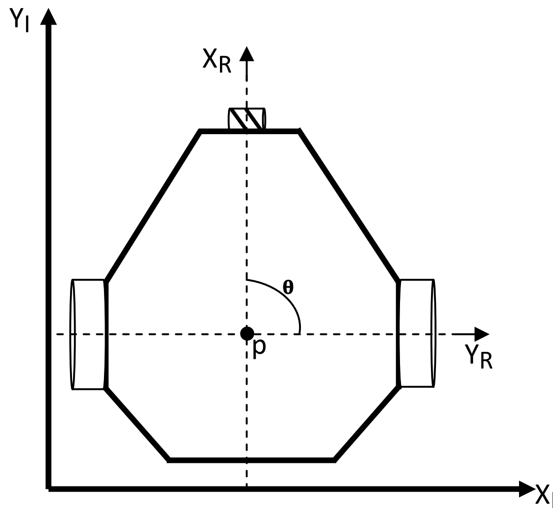


Fig 3.4. Differential mobile robot with two fixed wheels and one Swedish wheel

Figure 3.4 represents three wheeled differential mobile robot having two fixed standard wheels and one swedish wheel. For this type of robot, rank [C_{1f}] is one and it has no steerable standard wheels.

$$\delta_M = \delta_m + \delta_s \tag{3.20}$$

This results in the degree of mobility δ_m =2 and the degree of steerability δ_s =0,

The degree of maneuverability $\delta_M = \delta_m + \delta_s$ =2

3.8 HOLONOMICITY OF MOBILE ROBOT

In the robotics community, when describing the path space of a mobile robot, often the concept of holonomy is used. The term holonomy has broad applicability to several mathematical areas, including differential equations, functions and constraint expressions. In mobile robotics, the term refers specifically to the kinematic constraints of the robot chassis.

3.8.1 DIFFERENCE BETWEEN HOLONOMIC & NONHOLONOMIC

A nonholonomic kinematic constraint requires a differential relationship, for example the derivative of a position variable. Moreover, it cannot be integrated to provide a constraint in terms of the position variables only. A holonomic kinematic constraint can be expressed as an explicit function of position variables only. For example, in the case of a mobile robot with a single fixed standard wheel, a holonomic kinematic constraint would be expressible using α, β, l, r, □, x, y, $\boldsymbol{\theta}$ only. Such a constraint may not use derivatives of these values, such as □ or ξ.

- A nonholonomic mobile robot configuration is described by more than three coordinates. Three values are needed to describe the location and orientation of the robot, while others are needed to describe the internal geometry. However, a holonomic mobile robot can be described by three coordinates. The internal geometry does not appear in the kinematic equations of the abstract mobile robot, so it can be ignored. The robot can instantly develop a wrench or accelerate in an arbitrary combination of directions X, Y and $\boldsymbol{\theta}$.

- Nonholonomic robots are most prevalent because of their simple design and ease of control. By their nature, nonholonomic mobile robots have fewer degrees of freedom than holonomic mobile robots. These few actuated degrees of freedom in nonholonomic mobile robots are often independently controllable or mechanically decoupled, further simplifying the low-level control of the robot. Since they have fewer degrees of freedom, there are certain motions they cannot perform. This creates difficult problems for motion planning and implementation of reactive behaviors.

- Holonomicity offers full mobility with the same number of degrees of freedom as the environment. This makes path planning easier because there aren‘t constraints that need to be integrated. Implementing reactive behaviors is easy because there are no constraints which limit the directions in which the robot can accelerate.

- In case of nonholonomic mobile robot, the wheels rotate in the forward direction and then backward to its previous angular position, the robot will not necessarily arrive in the same location due to slippage or any other conditions.

- In cases of holonomic mobile robots, the wheels rotate in the forward direction and then backward to its previous angular position, the robot will arrive in the same location. So, holonomic robot can perform both Forward Kinematics (The angular rate difference between both wheels determines position & orientation of robot) and Inverse Kinematics (The position and orientation of a robot determines the angular rate difference between both wheels).

Considering equation (3.12), this constraint must use robot motion rather than pose because the point is to constrain robot motion perpendicular to the wheel plane to be zero. The constraint is non-integrable, depending explicitly on robot motion. Therefore, the sliding constraint is a nonholonomic constraint and the robot is a nonholonomic one.

3.9 KINEMATIC MODEL OF MOBILE ROBOT

The model of mobile robot (in Figure 3.5) consists of a vehicle chassis with two driving wheels mounted on the same axis and a front point sliding support. Both wheels have the same diameter denoted by 2r and separated by distance 2R. The two driving wheels are independently driven to achieve the motion and orientation. The kinematics of the differential drive mobile robot is based on the assumptions are as follows:

(1) Mobile robot moves on a plane surface.

(2) The wheel of a mobile robot rolls on the floor without translational slip.

(3) The wheel of a mobile robot makes rotational slip at the contact point between each wheel and the floor.

(4) The robot motion is slow such that the longitudinal traction & lateral force exerted on the robot‘s tires do not exceed the maximum static friction between tires and floor.

In Figure 3.5, Let x_c, y_c be the Cartesian coordinates of the point P in the middle of the rear axle respectively x_g, y_g the coordinates of the center of mass of the platform, the point G, and let $\boldsymbol{\theta}$ be the angle between the heading direction and the OX_I-axis specifying the orientation of the local platform with respect to the inertial frame. The distance between points G and C is '*d*'. The generalized coordinates $q_g=[\ x_g\ y_g\ \theta]^T$ or $q_c =[\ x_c\ y_c\ \theta]^T$ completely specifies the position of the robot in the X_IOY_I inertial Cartesian frame with a linear speed $v_c\left(\left[\dot{x}_c\ \dot{y}_c\right]^T\right)$ and angular velocity $\omega\left(\dot{\theta}\right)$.

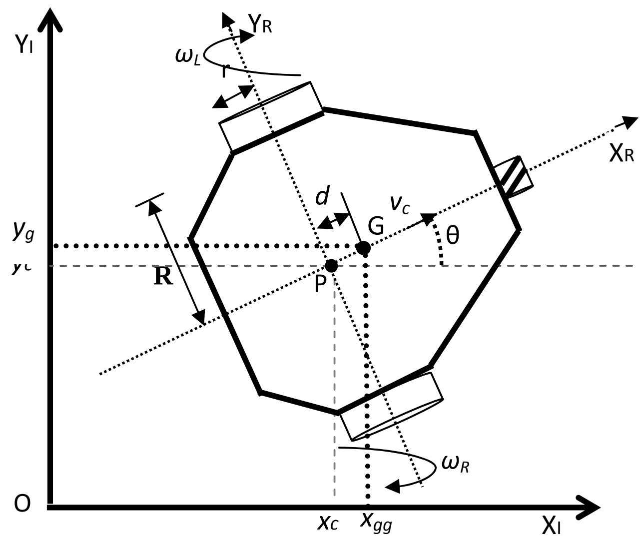


Fig. 3.5. Kinematic analysis of mobile robot

There three fundamental operations during kinematic motion [5]:

- ❖ If the angular velocities are identical both as values and relative senses $\omega_R = \omega_L$, the robot makes a linear motion. The direction on the linear motion, forward or backwards, depends of the opposite group of sense of the driven wheels angular velocities.

- ❖ If the angular velocities are identical as values but opposite as senses $\omega_R = -\omega_L$, the robot makes a -spin motion. The spin motion is a rotation of the mobile robot body around its vertical axis passing through the geometrical symmetry point (or centre of gravity). There is a particularity of this mechanical configuration, because only the two-wheeled differential drive mobile robot can do this type of motion, very useful to escape outside from difficult obstacles

- ❖ If the angular velocities are different as values and with the same senses, the robot makes a curve motion. Of course, the characteristics of the curve motion, i.e. the curvature coefficient *k* of the curve-segment trajectory, depend of the differences between the values of the two drive wheels. As the difference is smaller, as the curve motion tends to a linear motion

The kinematics of the differential drive mobile robot is based on the assumption of pure rolling and there is no slip between the wheel and surface

$$v_c = \frac{v_R + v_L}{2} \tag{3.21}$$

$$\omega = \frac{v_R - v_L}{2R} \tag{3.22}$$

Where $v_R = r\omega_R, v_L = r\omega_L$

So, in matrix form:

$$\begin{bmatrix} v_c \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2R} & \frac{-r}{2R} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \tag{3.23}$$

Suffix *R, L* and *t* stand for right, left wheel and tangential (with respect to its centre of gravity point of mobile robot) respectively.

From Figure 3.5, we can derive,

$$x_g = x_c + d\cos\theta \tag{3.24}$$

$$y_g = y_c + d\sin\theta \tag{3.25}$$

The linear velocity *vc* can be decomposed at point C in two components, as

$$\dot{x}_c = v_c \cos\theta \tag{3.26}$$

$$\dot{y}_c = v_c \sin\theta \tag{3.27}$$

So, the velocity components of *v*g at point G,

$$\dot{x}_g = v_c \cos\theta - d\omega \sin\theta \tag{3.28}$$

$$\dot{y}_g = v_c \sin\theta + d\omega \cos\theta \tag{3.29}$$

By eliminating *vc* from the equations, we can get a nonholonomic constraint:

$$\dot{x}_g \sin\theta - \dot{y}_g \cos\theta + d\dot{\theta} = 0 \tag{3.30}$$

$$\Rightarrow \begin{bmatrix} \sin\theta & -\cos\theta & d \end{bmatrix} \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = 0 \tag{3.31}$$

This relation states that the robot can only move in the direction normal to the axis of the driving wheels as long as mobile robot satisfies the conditions of pure rolling and non-slipping. Therefore, the component of the velocity of the contact point with the ground, orthogonal to the plane of the wheel is zero.

When the center of mass of the platform, the point G, coincides with its center of rotation, the point C, then d=0, so nonholonomic constraint will be:

$$\dot{x}_g \sin\theta - \dot{y}_g \cos\theta + 0 = 0 \tag{3.32}$$

Combining linear and angular velocities at point G (from equation 3.26 and 3.27) can be written in matrix form,

$$\dot{q} = \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega \end{bmatrix} \tag{3.33}$$

According to equations (3.21) and (3.31), the kinematic model of differential drive two wheeled mobile robot can be explicitly written as:

$$\dot{q} = \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta - \frac{r}{2R} d\sin\theta & \frac{r}{2}\cos\theta + \frac{r}{2R} d\sin\theta \\ \frac{r}{2}\sin\theta + \frac{r}{2R} d\cos\theta & \frac{r}{2}\sin\theta - \frac{r}{2R} d\cos\theta \\ \frac{r}{2R} & -\frac{r}{2R} \end{bmatrix} \begin{bmatrix} v_c \\ \omega \end{bmatrix} \tag{3.34}$$

3.10 CONCLUSION

When developing a robot, the designer analyzes the terrain in which the robot will be travelling and what the function of the robot on reaching the goal. With the help of developed methodology, the robot can achieve path following considering both kinematic model of the mobile robot. According to this analysis the robot’s locomotion mechanism can be chosen.

CHAPTER 4

FUZZY LOGIC CONTROLLER FOR MOBILE ROBOT

4. FUZZY LOGIC CONTROLLER FOR MOBILE ROBOT

An autonomous mobile robot is a machine that operates in an unknown and unpredictable environment. Uncertainty and ambiguity associated with reactive navigation for autonomous mobile agent in unknown or partially known chaotic surroundings, especially, unpredictably changing environment can be unravelled by making coordination and fusion of the elementary behaviours of mobile agent. A key issue in the research of an autonomous mobile robot is the design and development of a control technique that enables the robot to navigate in a real world environment, avoiding structured and unstructured obstacles especially in crowded and unpredictably changing environment. The fuzzy navigation technique, which is accomplished to generate satisfactory direction and velocity manoeuvers of the autonomous robot, is instigated here for the robot navigation to reach its goal safely moving on unknown static terrains. This chapter presents the development in the area of intelligent controller for mobile robot in various (known and unknown) environments where action coordination of the behaviours will be addressed using fuzzy logic in the present research. The inputs to the proposed fuzzy control scheme consist of a target angle between a robot and a specified target and the distances between the robot and the obstacles to the left, front, and right to its locations, being acquired by an array of sensors. In this chapter an intelligent controller has been proposed for mobile robot navigation algorithm employing fuzzy theory in a complex environment.

4.1 INTRODUCTION

Human beings do not need precise, numerical information input to make a decision, but they are able to perform highly adaptive control. Humans have the remarkable capability of performing a wide variety of physical and mental tasks without any explicit measurements or computations. Examples of everyday tasks being parking a car, driving in city traffic, playing golf or summarizing a story. In performing such familiar tasks, humans use perceptions of time, distance, speed, shape, and other attributes of physical and mental objects [149]. Fuzzy logic is a problem-solving control system methodology that lends itself for implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, workstation-based data acquisition and control systems. The theory of fuzzy logic systems is inspired by the remarkable human capability to operate on and reason with perception-based information. The rule-based fuzzy logic provides a scientific formalism for reasoning and decision making with uncertain and imprecise information. It can be implemented in hardware, software, or a combination of both. Fuzzy logic approach to control problems mimics how a person would make decisions. The main advantages of a fuzzy navigation strategy lie in the ability to extract heuristic rules from human experience, and to obviate the need for an analytical model of the process [54, 68].

Fuzzy logic technique is having ability to take decision like human being, avoidance of structured and unstructured obstacle in complex environment. The FLC (fuzzy logic controller) is a problem-solving control system that attends to the implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems [85].The rule-based fuzzy logic provides a scientific formalism for reasoning and decision making with uncertain and imprecise information. It can be implemented in hardware, software, or a combination of both. Fuzzy logic approach to control problems mimics how a person would make decisions. Another advantage of fuzzy navigation is its ability to extract heuristic rules from human experience, and to preclude the need for an analytical model of the process [86].

Fuzzy controllers consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value. The processing stage is based on a collection of logic rules in the form of IF-THEN statements, where the IF part is called the "antecedent" and the THEN part is called the "consequent". The fuzzy rule sets have several antecedents that are combined using fuzzy operators like AND, OR, and NOT where AND uses the minimum weight of all the antecedents, OR uses the maximum value and NOT operator subtracts a membership function from 1 to give the "complementary" function [87]. The complete process of applying fuzzy logic to a problem can be seen in fig1. Here the crisp space values are the real world inputs to the problem.

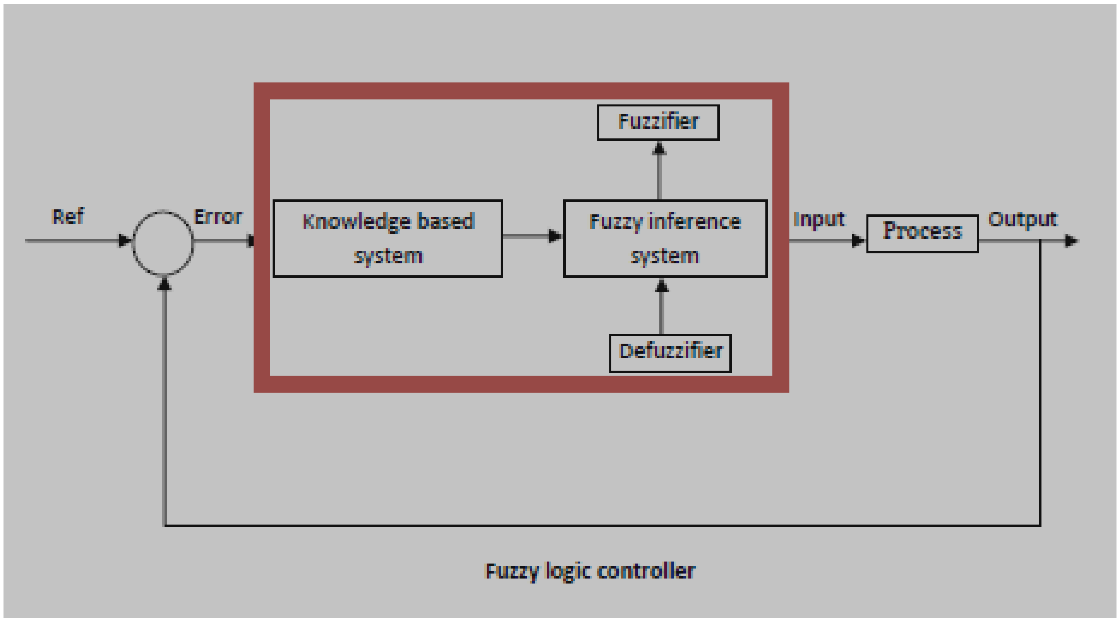


Fig. 4.1. Process for the working of a fuzzy logic controller

Recent researches have shown many advantages of the fuzzy based evolutionary navigation scheme over most other techniques (such as potential field method, vector field histogram and local navigation etc.), one of them being that less local information is required for this algorithm. Fuzzy logic can be used in implement individual behaviours, to coordinate the various behaviours, to select roles for each robot, and for robot perception, decision-making, and speed control [150]. Fuzzy behaviour-based architecture for mobile robot navigation in unknown environments incorporates design of basic behaviours for mobile robot navigation: goal seeking behaviour, obstacle avoidance behaviour, wall following behaviour, and deadlock disarming behaviour [151, 152]. Each of the behaviours is implemented using fuzzy controller to achieve the oriented navigation task.

The development of techniques for autonomous navigation in real-world environments constitutes one of the major trends in the current research on robotics. One of the important problems of autonomous mobile robot navigation is the need to cope with the large amount of uncertainty that is inherent of natural environments. Fuzzy logic has features that make it an adequate tool to address this problem. Navigation of mobile robots in presence of static and moving obstacles using fuzzy technique is presented in this work. At first, a set of navigation rules are extracted from the data base. The rules are used to control the navigation of mobile robots. The use of fuzzy logic techniques for controlling wheel-based mobile robots has been effectively proposed by many authors in the last decade [153, 67, 60, 154]. This chapter proposes an on-line path analysis and planning approach that embeds a fuzzy strategy to drive a mobile robot. A new intelligent fuzzy interface system has been developed in this current investigation. In this approach, the fuzzy logic system is used to control the robot taking inputs from various sensors. Sensor signals are fed to the fuzzy logic system, and the output provides motor control commands (e.g., turn right or left). The fuzzy logic system learns the full dynamics of the mobile robot online. Fuzzy controller for mobile robot has four inputs and two outputs. Both inputs and output have three membership functions. Each membership function consists of trapezoidal and triangular membership functions. In this methodology 81 rules have been used to design the fuzzy controller. This research focuses a fuzzy logic framework to be implemented in the mobile robot for behaviour design and coordination. The proposed method has been compared with other methods [153, 73, 154, 155] which show the effectiveness of the developed method. It is also concluded that the current method can be successfully employed for navigation of mobile robot. This fuzzy controller of mobile robot for path analysis and planning has been authenticated by experimental verification.

4.2 FUZZY LOGIC CONTROLLER

Fuzzy logic controllers work by mimicking the ability of human brain to take decisions. Hence it can be formulated using a simple set of rules called fuzzy rules also known as If-Then rules. It consists of inputs mentioned as If condition or antecedent and the output mentioned as then condition or consequence. This type of input-output coupling is easily understandable and consists of natural linguistic representations.

A fuzzy inference system is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules and fuzzy reasoning. It mainly consists of three main components namely a rule base which contains a selection of fuzzy rules, a database that defines the membership function used in fuzzy rules and a reasoning mechanism that performs the inference procedure. It can take either fuzzy inputs or crisp inputs but the outputs are almost always fuzzy sets. Sometimes it is important to have a crisp output mainly in a situation where a fuzzy inference system is used as a controller. So a defuzzification method is applied to extract a crisp value in order to represent a fuzzy set.

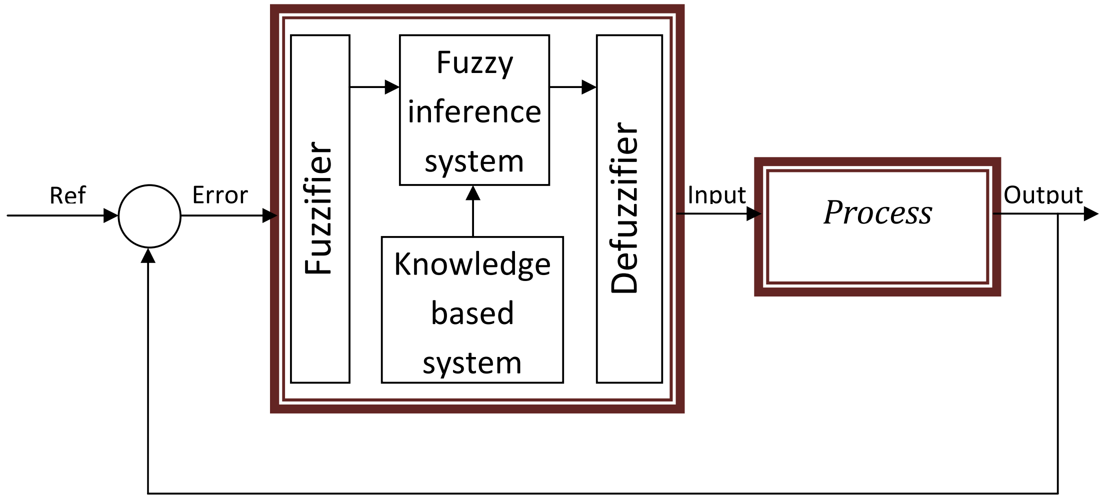


Fig. 4.2. Fuzzy controller with fuzzy inference system

4.3 MEMBERSHIP FUNCTION

The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, it represents the degree of truth as an extension of valuation. Degrees of truth are often confused with probabilities, although they are conceptually distinct, because fuzzy truth represents membership in vaguely defined sets, not likelihood of some event or condition. The membership function can have varying shapes depending on the type of variations in input and output. For example, a triangular membership function is used when we need a sharp value while a trapezoidal function is used when we need a constant value.

For any set X, a membership function on X is any function from X to the real unit interval [0, 1].Membership functions on X represent fuzzy subsets of X. The membership function which represents a fuzzy set $\tilde{A}$ is usually denoted by μ_A. For an element x of X, the value μ_A (x) is called the membership degree of x in the fuzzy set $\tilde{A}$ The membership degree μ_A (x) quantifies the grade of membership of the element x to the fuzzy set $\tilde{A}$ The value 0 means that x is not a member of the fuzzy set; the value 1 means that x is fully a member of the fuzzy set The values between 0 and 1 characterize fuzzy members, which belong to the fuzzy set only partially.

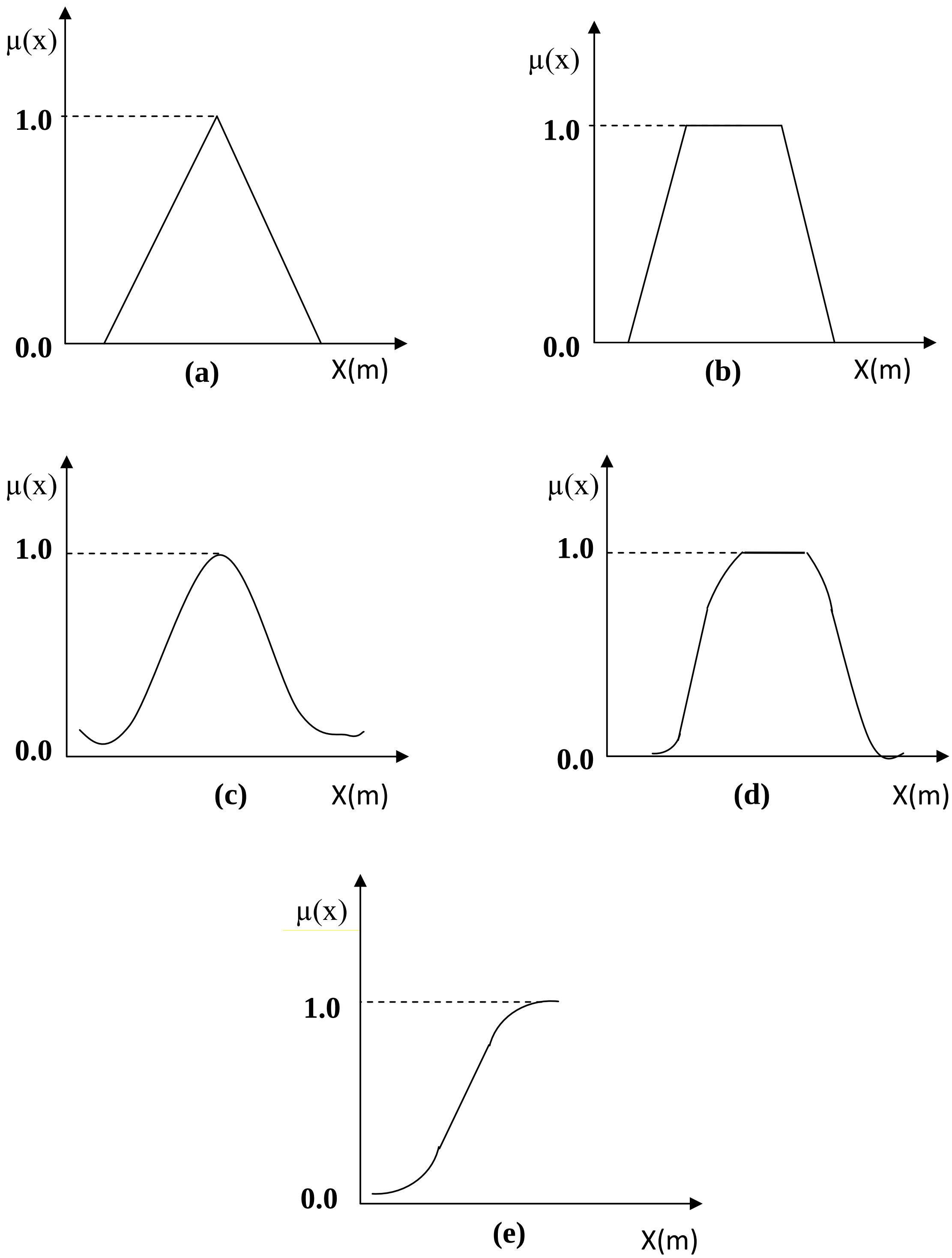


Fig. 4.3. Types of membership function: (a) triangular (b) trapezoidal (c) gaussian (d) bell shaped (e) sigmoidal

Sometimes, a more general definition is used, where membership functions take values in an arbitrary fixed algebra or structure L; usually it is required that L be at least a poset or lattice. The usual membership functions with values in [0, 1] are then called [0, 1]-valued membership functions. Elements of a fuzzy set are taken from a universe of discourse or just universe. The universe contains all elements that can come into consideration. Before designing the membership functions it is necessary to consider the universes for the inputs and outputs.

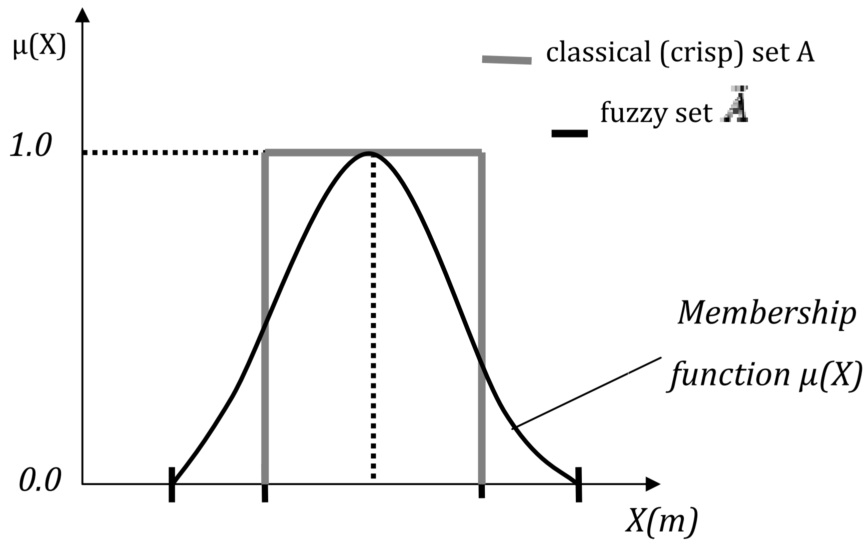


Fig. 4.4. Example of a membership function.

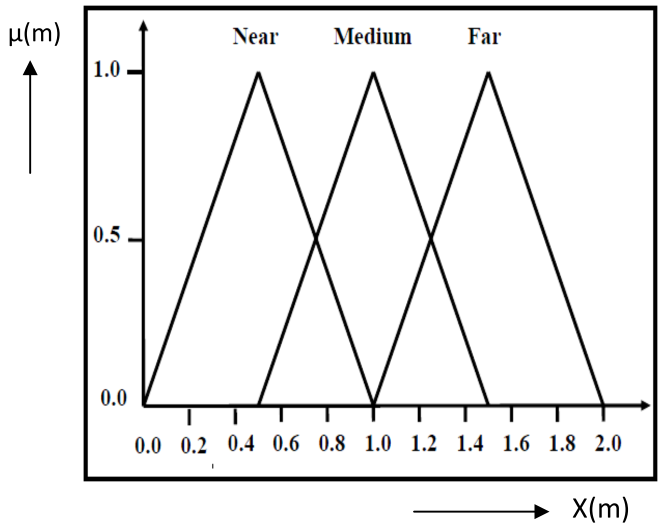
4.4 FUZZY CONTROLLER FOR THE MOBILE ROBOT

Fuzzy controllers are very simple conceptually. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value.

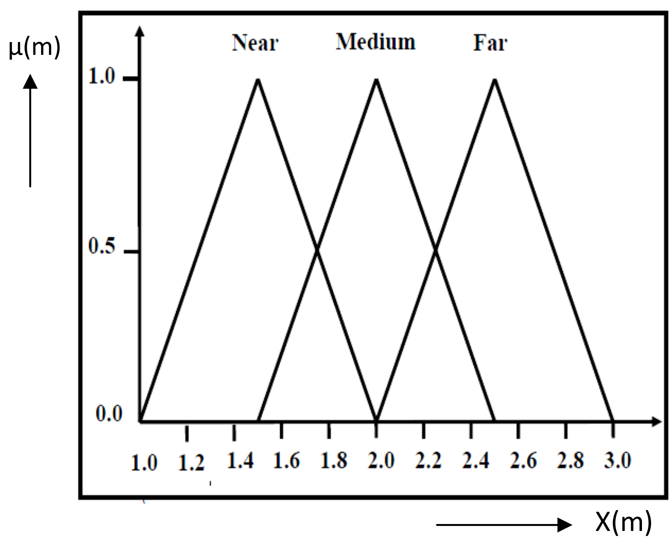
For the given robot, the fuzzy controller has three inputs and two outputs. The inputs are given as distances whereas the outputs are received as velocities. The inputs given to the fuzzy controller are Left Obstacle Distance (LD), Right Obstacle Distance (RD), Front Obstacle Distance (FD) and Steering Angle (STA). Similarly, the outputs received are Left Wheel Velocity (LWV) and Right Wheel Velocity (RWV).

4.4.1 MEMBERSHIP FUNCTIONS FOR THE ROBOT CONTROLLER

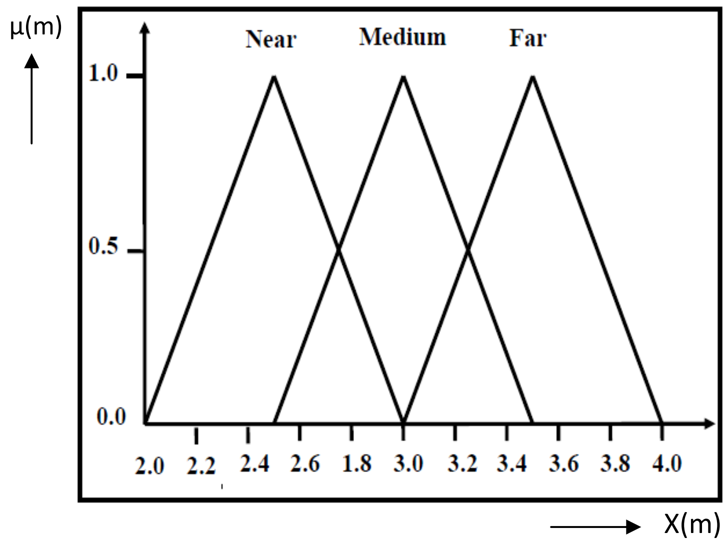
The membership function for each of the four input variables are derived using the sets of operator input/output data. For the sake of ease in application and understanding, the membership function considered is triangular. The membership functions for the inputs and outputs along with the range of these functions are shown in the figure below.



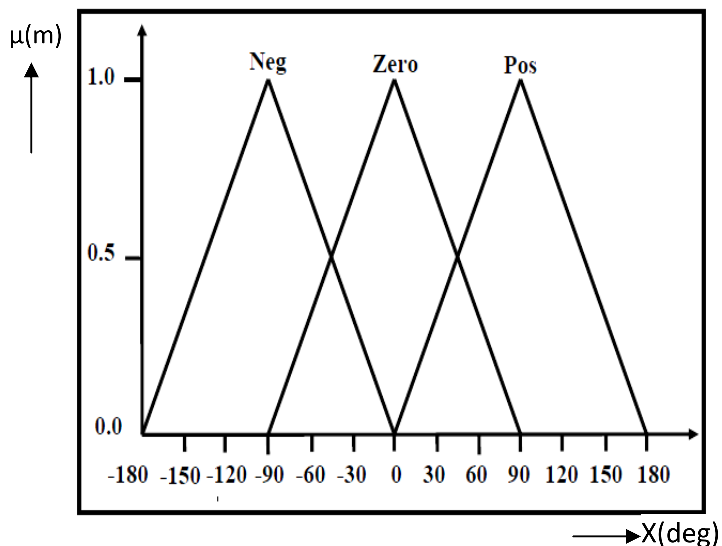
Membership function for FD



Membership function for LD



Membership function for RD



Membership function for STA

Fig. 4.5. All In put membership functions for fuzzy controller

Three membership functions are taken for each input variable but their ranges vary from each other. For front obstacle distance, left obstacle distance and right obstacle distance, the ranges taken are 0 to 2, 1 to 3 and 2 to 4 respectively and the membership functions taken are near, medium and far. Each function again has a range between which it varies. For steering angle the membership functions taken are positive, zero and negative. The range of the membership functions lies between the ranges of -180 to 180.

4.4.2 FUZZIFICATION

It is the conversion of crisp input data to degrees of membership by mapping data from crisp input space to the fuzzy sets. The fuzzy set, here, is labeled by linguistic variables near, medium and far and these are chosen to fuzzily left obstacle distance, right obstacle distance and front obstacle distance. Variables “pos”, “zero” and “neg” define the heading angle of the target with respect to the robot.

When the target is located at the left side of the mobile robot, the target angle is considered negative and if the target is at right side of robot, the target angle is considered positive. The term “no target considered” is used if there is no target in the environment. Linguistic variables like “fast”, “medium” and “slow” are defined for left wheel velocity and right wheel velocity for three membership function. The parameters for Left Obstacle Distance, Front Obstacle Distance, Right Obstacle Distance, Steering Angle, Left Wheel Velocity and Right Wheel Velocity are given in table 4.1, 4.2 and 4.3 respectively.

Variables	Near (Meter)	Medium (Meter)	Far (Meter)
Left Obstacle Distance	0.0	1.0	2.0
Right Obstacle Distance	1.0	2.0	3.0
Front Obstacle Distance	2.0	3.0	4.0

Table 4.1. Parameters for left, right and front obstacle distance

Variable	Negative (degree)	Zero (degree)	Positive (degree)
Heading angle	-180	-90	0
	-90	0	90
	0	90	180

Table 4.2. Parameters for Steering angle

Variables	Slow (meter/sec)	Medium (meter/sec)	Fast (meter/sec)
Left wheel velocity	0.0	0.5	1.0
Right wheel velocity	0.5	1.0	1.5

Table 4.3. Parameters for Left and right wheel velocity

4.5 FUZZY RULE MECHANISM

Fuzzy rules are formulated based on human perception. The fuzzy rule base is a set of linguistic rules in the form of "if a set of conditions are satisfied, then a set of consequences are inferred". Based on the above fuzzy subsets, the fuzzy control rules are defined in a general form for four inputs and two outputs fuzzy system as follows:

If (matching degree of LD is $\mu(LD_i)$ and matching degree of FD is $\mu(FD_j)$ and matching degree of RD is $\mu(RD_k)$ and matching degree of HA is $\mu(STA_m)$, Then (matching degree of LWV is $\mu(LWV_{ijkm})$ and matching degree of RWV is $\mu(RWV_{ijkm})$. (4.1) where i = 1 to 3, j = 1 to 3, k = 1 to 3 and m =1 to 3 because LD, FD, RD and STA have three membership functions each.

The matching degree of final output is computed by the following formula:

Matching degree $\mu_{LWV, RWV}(vel_{ijkm}) = \min\{\mu(LD_i), \mu(FD_j), \mu(RD_k) \text{ and } \mu(STA_m)\}$ (4.2)

When the matching degree=1 the inferred conclusion is identical to the rule's consequent, and if it is zero no conclusion can be inferred from the rule.

Finally, the output firing area of the left and right wheel velocities can be computed by the following formula,

$$\left.\begin{aligned} \mu_{LV}(vel) &= \max\{\mu_{LV}(vel1111), \ldots\ldots\ldots\ldots\mu(vel_{ijkm}), \ldots\ldots\ldots\ldots\mu(vel\ 3333)\} \\ \mu_{RV}(vel) &= \max\{\mu_{RV}(vel\ 1111), \ldots\ldots\ldots\ldots\mu(vel_{ijkm}), \ldots\ldots\ldots\ldots\mu(vel\ 3333)\} \end{aligned}\right\} \quad (4.3)$$

The final output (crisp value) of the fuzzy logic controller of left and right wheel velocities can be calculated by "Centre of Gravity" method.

$$\begin{cases} Left\ velocity = LV = \dfrac{\int vel\ .\mu_{LV}(vel)\ .d(vel)}{\int \mu_{LV}(vel)\ .d(vel)} \\ Right\ velocity = RV = \dfrac{\int vel\ .\mu_{RV}(vel)\ .d(vel)}{\int \mu_{RV}(vel)\ .d(vel)} \end{cases} \quad (4.4)$$

4.5.1 SIMULATION

The conditions given to the robot as input are constructed in the form of rules which are then given to the simulator to find out the required output. The rules are in the form of "if – then" as mentioned earlier for fuzzy rules. Since there are four inputs and two outputs hence the total number of rules formulated can range upto 256. Table 4.1 lists some of the input conditions and the outputs obtained from them. The front obstacle distance, left obstacle distance and right obstacle distance have three values each, i.e. near, medium and far while the steering angle has positive, negative and zero as its three values. The output variables are the velocities of the left and right wheels. Hence the values for them are slow, medium and fast.

Sl. No.	FOD	LOD	ROD	STA	LWV	RWV
1	Near	Near	Far	Positive	Slow	Fast
2	Medium	Near	Far	Zero	Medium	Medium
3	Far	Near	Far	Negative	Fast	Slow
4	Near	Near	Medium	Positive	Slow	Fast
5	Medium	Near	Medium	Zero	Medium	Medium
6	Far	Near	Medium	Negative	Fast	Slow
7	Near	Near	Near	Positive	Slow	Fast
8	Medium	Near	Near	Zero	Medium	Medium
9	Far	Near	Near	Negative	Fast	Slow
10	Near	Medium	Far	Positive	Slow	Fast
11	Medium	Medium	Far	Zero	Medium	Medium
12	Far	Medium	Far	Negative	Fast	Slow
13	Near	Medium	Near	Positive	Slow	Fast
14	Medium	Medium	Near	Zero	Medium	Medium
15	Far	Medium	Near	Negative	Fast	Slow
16	Near	Medium	Medium	Positive	Slow	Fast
17	Medium	Medium	Medium	Zero	Medium	Medium
18	Far	Medium	Medium	Negative	Fast	Slow
19	Near	Far	Far	Positive	Slow	Fast
20	Medium	Far	Far	Zero	Medium	Medium
21	Far	Far	Far	Negative	Fast	Slow
22	Near	Far	Medium	Positive	Slow	Fast
23	Medium	Far	Medium	Zero	Medium	Medium
24	Far	Far	Medium	Negative	Fast	Slow

25	Near	Far	Near	Positive	Slow	Fast
26	Medium	Far	Near	Zero	Medium	Medium
27	Far	Far	Near	Zero	Medium	Medium
28	Near	Near	Far	Negative	Fast	Slow
29	Medium	Near	Far	Zero	Medium	Medium
30	Far	Near	Far	Positive	Slow	Fast
31	Near	Near	Medium	Negative	Fast	Slow
32	Medium	Near	Medium	Zero	Medium	Medium
33	Far	Near	Medium	Positive	Slow	Fast
34	Near	Near	Near	Negative	Fast	Slow
35	Medium	Near	Near	Zero	Medium	Medium
36	Far	Near	Near	Positive	Slow	Fast
37	Near	Medium	Far	Negative	Fast	Slow
38	Medium	Medium	Far	Zero	Medium	Medium
39	Far	Medium	Far	Positive	Slow	Fast
40	Near	Medium	Far	Negative	Fast	Slow
41	Medium	Medium	Medium	Zero	Medium	Medium
42	Far	Medium	Medium	Negative	Fast	Slow
43	Near	Far	Near	Negative	Fast	Slow
44	Medium	Far	Near	Positive	Medium	Medium
45	Far	Far	Near	Positive	Slow	Fast

Table 4.4. Some of the input and output conditions

These values were given to the MATLAB simulator and a program for the navigation of the robot was created. The microchip in the robot was instilled with the program written in the MATLAB simulator. Various obstacles were provided in the path of the robot so that it had to change its path to reach its destination. The figure showing the arrangement of the obstacles is shown in the figure 4.3.

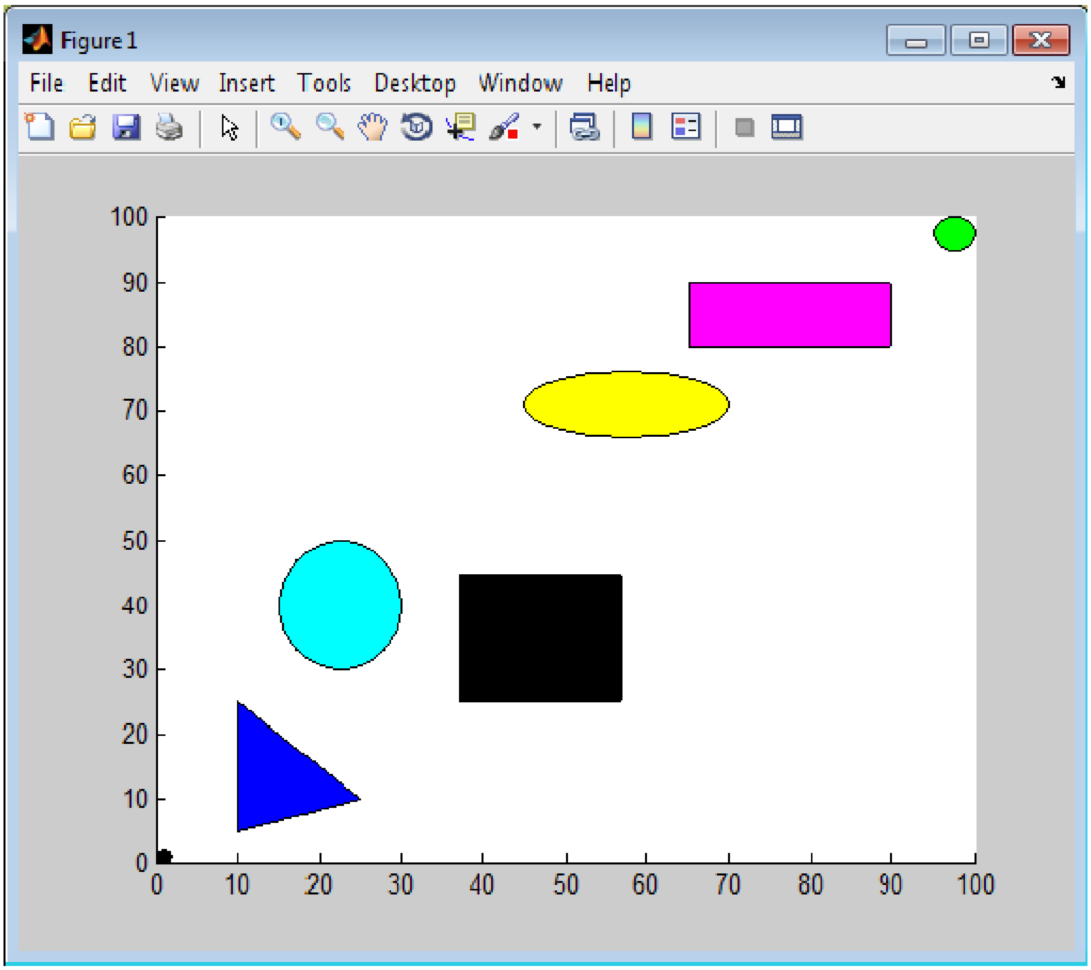


Fig. 4.6. Figure showing the obstacles posed in the robots path

The robot can be seen as a small black dot at the origin of the graph whereas the destination is seen as the green circle at the other end of the graph. The robot has to dodge the obstacles and reach the destination. The program allows the robot to reach the destination safely without colliding with the obstacles. Figure 4.4 shows the path of the robot taken to reach the destination.

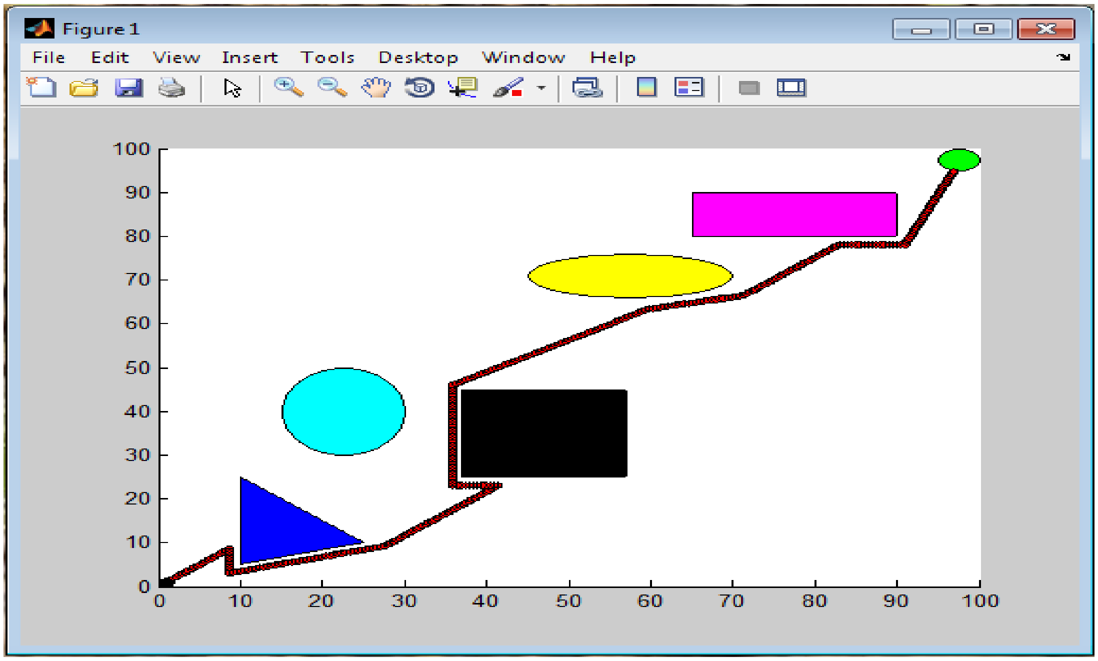


Fig. 4.7. Path taken by the robot during simulation in MATLAB for Fuzzy logic

4.6 EXPERIMENTAL ANALYSIS

The path achieved by the robot during simulation was verified by implementing the program on a stingray robot in a physical environment similar to the environment produced in simulation. The path length of the robot was measured in the real environment as well as in the simulation.

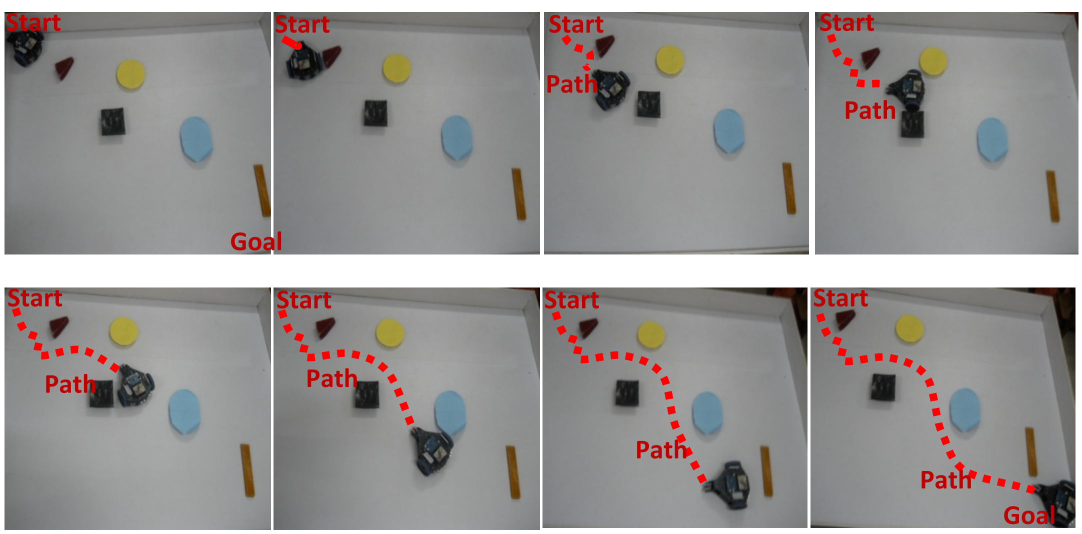


Fig. 4.8. Path taken by the robot during experiment using Fuzzy controller

4.7 RESULTS AND DISCUSSION

The comparison between the simulation and experimental results was done in terms of the distance covered by the robot and the time taken. The comparison for Fuzzy logic is given in the following table:

Sl. No .	Algorithm used to find the robot's path	Path length in simulation achieved by proposed technique in cm.	Path length in experiment achieved by proposed technique in cm.	Error in %
1.	Fuzzy logic controller	277.7	278.0	0.108

Table 4.5. Results found from the simulation and experimental results for Fuzzy Method

Scale for simulation: 1 = 27.777

4.8 CONCLUSION

The simulation of the above techniques in MATLAB revealed that the path taken by the robot in all situations is different. However the robot was successful in avoiding the obstacles in the path and was successful in reaching the goal position. The comparison between simulation and experimental results showed a good agreement.

CHAPTER 5

TYPE 2 FUZZY LOGIC FOR MOBILE ROBOT

5. TYPE 2 FUZZY LOGIC FOR MOBILE ROBOT

Fuzzy systems enable a machine to take decisions as the human mind would take. However, even human decision making process is not flawless and sometimes wrong decisions hamper the navigation to the goal. In such cases, improving the decision making process becomes a necessity. In order to do that, we have to reduce the ambiguities present in the decision making process. So we implement type 2 fuzzy system that would reduce anomalies and improve the decision making process. The present chapter gives us an overview of such process.

5.1 INTRODUCTION

Artificial Intelligence (AI) based systems work as gizmo that not just enhance human decision making but also compensate intrinsic flaws in human decision making processes. For qualitative and holistic development, it is important to take productive decisions and selections. In order to increase productivity and effectiveness of the selection process, it is essential to have an advisory system that can offer advantages of knowledge based approach. The database-oriented systems are comparatively less effective and offer necessary static decision support. Generalized systems might not be helpful in providing personalized assistance and effective advisory to an individual user. Hence, the scope of such systems is restricted and they are not widely accepted. Moreover, the knowledge based systems can provide more effective evaluation of suitability of decisions and learns from feedback.

Type-2 fuzzy systems generalize type-1 fuzzy sets and systems so that more uncertainty can be avoided. From the very beginning of fuzzy sets, criticism has been made about the fact that the membership function of a type-1 fuzzy set has no uncertainty associated with it. This contradicts the word fuzzy, as it has the possibilities of lots of uncertainties. Prof. L.A. Zadeh [161], proposed a more sophisticated kind of fuzzy set, the first of which was called a type-2 fuzzy set. A type-2 fuzzy set allows us to accommodate uncertainty about the membership function into fuzzy set theory, and creates a way to address the criticism of type-1 fuzzy sets head-on. And, if there is no uncertainty, then a type-2 fuzzy set reduces to a type-1 fuzzy set that is analogous to probability reducing to determinism when unpredictability vanishes.

Type-1 Fuzzy logic systems have limited capacity to handle data uncertainties [162]. Once a type-1 membership function has been defined, uncertainty disappears because a type 1 membership function is precise [163]. Type-2 fuzzy logic systems make it possible to handle uncertainties in a better way. These are rule based systems where linguistic variables are defined by means of type-2 fuzzy sets that include a footprint of uncertainty (FOU) [4]. It captures more uncertainties [162] than type 1 fuzzy set systems. Type 2 fuzzy sets have non-crisp membership functions whereas type 1 fuzzy sets have crisp membership grades [165]. A representation of the inference model for type 2 fuzzy set systems is depicted in Figure 1 [162].

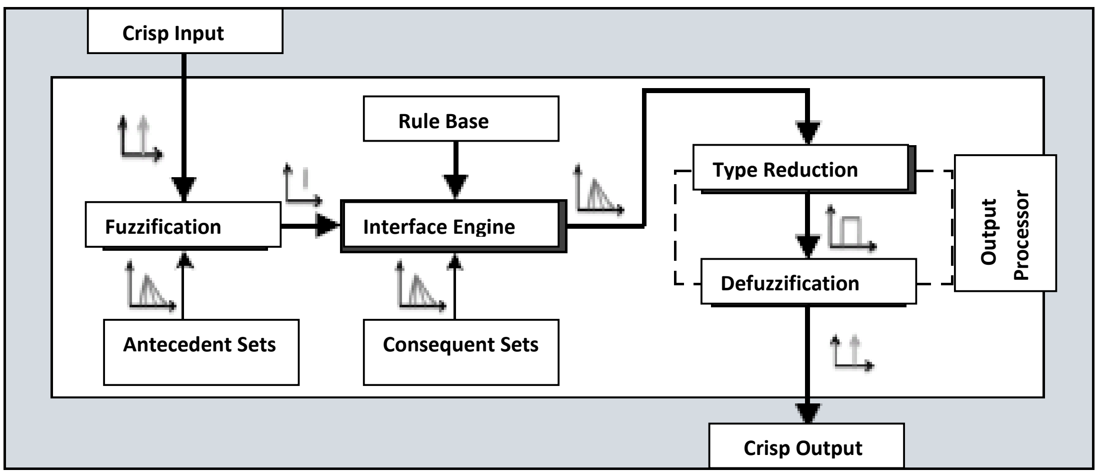


Fig. 5.1. Inference model for type 2 fuzzy set systems

The process begins with fuzzification, which maps crisp points into type 2 fuzzy set systems. After that the inference engine computes the rule base by making logical combinations of antecedent type 2 fuzzy set systems, whose results are concerned with consequent type 2 fuzzy systems to form an aggregate output type 2 fuzzy set. Then type reduction (TR) takes all the output sets and performs a centroid calculation of this combined type 2 fuzzy set, and this leads to a type 1 fuzzy set called type reduced set. That reduced set is finally defuzzified in order to obtain a crisp output [164, 166]. The computational complexity of this model is reduced if interval type 2 fuzzy sets are used [164]. It is convenient in the context of hardware implementation in order to make softer the computational effort and speed up the inference time. Type 2 fuzzy hardware is a topic of special interest, since the application of type 2 fuzzy logic system to particular fields that demand mobile electronic solutions would be necessary. Some recent applications of type 2 fuzzy logic systems have been developed in fields like robotics, communication and control systems among others [167, 163, 168, 169, 170]. It is worth to think about the possibility of embedding type 2 fuzzy logic systems handling these applications in order to achieve better communication speeds in smaller areas [178].

Control and robotics are one of the most widely used application fields of fuzzy logic. The advantages of type-2 fuzzy logic controllers over type-1 fuzzy logic controllers (T1-FLC) have also been demonstrated and documented in [167, 172, 173, 163, 174, 175]. Although this kind of works presents improvements in the type 2 fuzzy logic controllers performance, it is necessary to propose methodologies where these advances can be reflected in design processes as [172, 173,174,175] and hardware implementation approaches over embedded devices as [172, 176, 168, 169]. In this way type 2 fuzzy logic systems would become widely used in different applicative contexts.

5.2 INTERVAL TYPE 2 FUZZY SYSTEMS

Type 1 fuzzy system theory was first introduced by Zadeh in 1965 and has since been successfully applied in many areas, including modelling and control, data mining, time-series prediction, linguistic summarization, computing with words, etc.

A type 1 fuzzy system X is comprised of a domain D_X of real numbers (also called the universe of discourse of X) together with a membership function (MF) $\mu_X : D_X \rightarrow [0,1]$, i.e.,

$$X = \int_{D_X} \mu(x) / x$$

Here $\int$ denotes the collection of all points $x \in D_X$ with associated *membership grade* $\mu_X(x)$. In spite of having a name that represents uncertainty, research has shown that there are limitations in the ability of type 1 fuzzy system to model and minimize the effect of uncertainties. This is because a type 1 fuzzy system is certain in the sense that its membership grades are crisp values. Recently, type-2 fuzzy systems, characterized by membership functions that are themselves fuzzy, have been attracting great interests. Interval type 2 fuzzy systems, a special case of type-2 fuzzy systems, are currently the most widely used for their reduced computational cost, and are also the focus of this paper. The membership function for the interval type 2 fuzzy logic is shown in fig 5.3

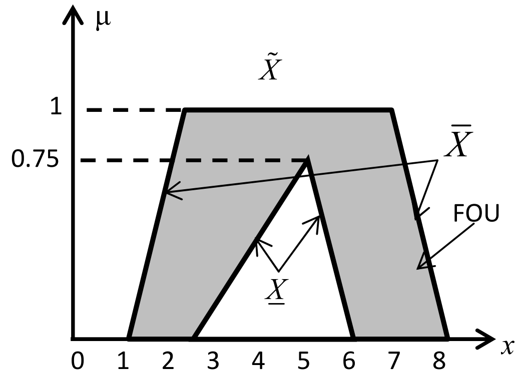


Fig 5.2 Membership function for interval type fuzzy logic

The membership function is not just a value but a range or interval unlike type 1 fuzzy controller. The upper limit of the membership function is given by $\overline{X}$ whereas the lower limit is given by $\underline{X}$. The area between both these limits is called the footprint of uncertainty or FOU.

5.3 INTERVAL TYPE 2 FUZZY LOGIC CONTROLLERS

The schematic diagram of an interval type 2 fuzzy logic controller is similar to its type 1 counterpart. The major difference is that at least one of the fuzzy systems in the rule base is an interval type 2 fuzzy system. Hence, the outputs of the inference engine are interval type 2 fuzzy systems, and a type reducer is needed to convert them into a type 1 fuzzy system before defuzzification can be carried out. The inputs given to the controller are left obstacle distance (LD), left obstacle distance (RD), front obstacle distance(FD) and steering angle(STA) whereas left wheel velocity(LWV) and right wheel velocity(RWV). Here the computation for one of the inputs LD is shown. The same process is repeated for all the inputs to receive the output.

$$R^n : IF\ x_1\ is\ \tilde{X}_1^n\ and \ldots x_I\ is\ \tilde{X}_I^n, THEN\ y\ is\ Y^n \qquad n = 1, 2, \ldots N$$

where $\tilde{X}_1^n$ (i = 1, . . . , I) are IT2 FSs and $Y^n = [\overline{y}^n, \underline{y}^n]$ is an interval, which can be understood as the centroid of a consequent interval type 2 fuzzy logic system.

Assume the input vector is LD′ = (LD1, LD2, ..., LD'I). Typical computations in an interval type 2 fuzzy logic system involve the following steps:

1. Compute the membership of LD_i on each LDi = 1, 2, ..., I, n = 1, 2, ...,N. (8)

2. Compute the firing interval of the nth rule, F^n(LD)using the formula:

$$F^n(x') = \left[\mu_{\underline{X}_1^n}(x'_1) \times \ldots \times \mu_{\underline{X}_I^n}(x'_I), \mu_{\bar{X}_1^n}(x'_1) \times \ldots \times \mu_{\bar{X}_I^n}(x'_I)\right] = \left[\underline{f}^n, \bar{f}^n\right], n = 1, 2, \ldots N \quad (9)$$

3. Perform type-reduction to combine F^n(LD) and the corresponding rule consequents. There are many such methods. The most commonly used one is the center-of-sets type-reducer whose formula is given by

$$Y_{cos}(x') = \bigcup_{\substack{f^n \in F^n(x') \\ y^n \in Y^n}} \frac{\sum_{n=1}^{N} f^n y^n}{\sum_{n=1}^{N} f^n} = [y_l, y_r] \quad (10)$$

The flowchart for the type 2 fuzzy system is shown in the next figure.

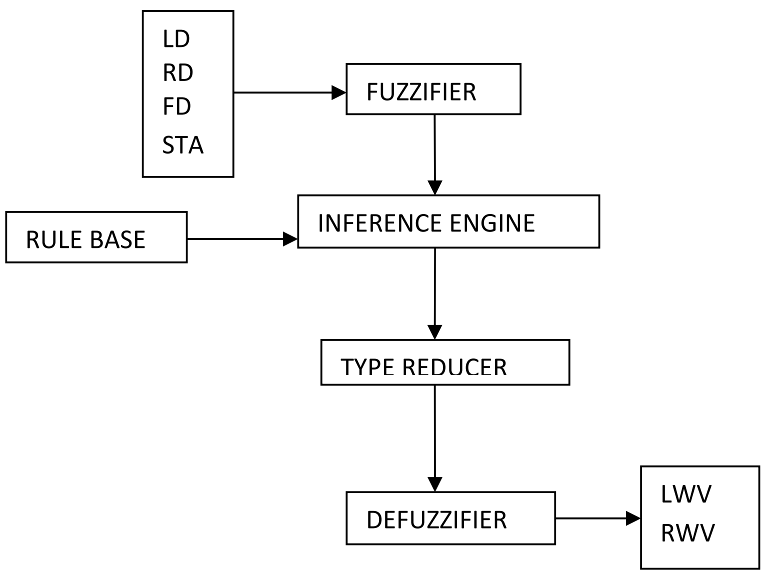


Fig 5.3 Flowchart for type 2 fuzzy controller for the given mobile robot

5.4 SIMULATION

The robot was given a set of instructions to follow and the program was written in MATLAB simulator. The microchip in the robot was instilled with the program written in the

MATLAB simulator. Various obstacles were provided in the path of the robot so that it had to change its path to reach its destination. The figure showing the arrangement of the obstacles is shown in the figure 5.3. The robot can be seen as a small black dot at the origin of the graph whereas the destination is seen as the green circle at the other end of the graph. The robot has to dodge the obstacles and reach the destination. The program allows the robot to reach the destination safely without colliding with the obstacles. Figure 5.4 shows the path of the robot taken to reach the destination.

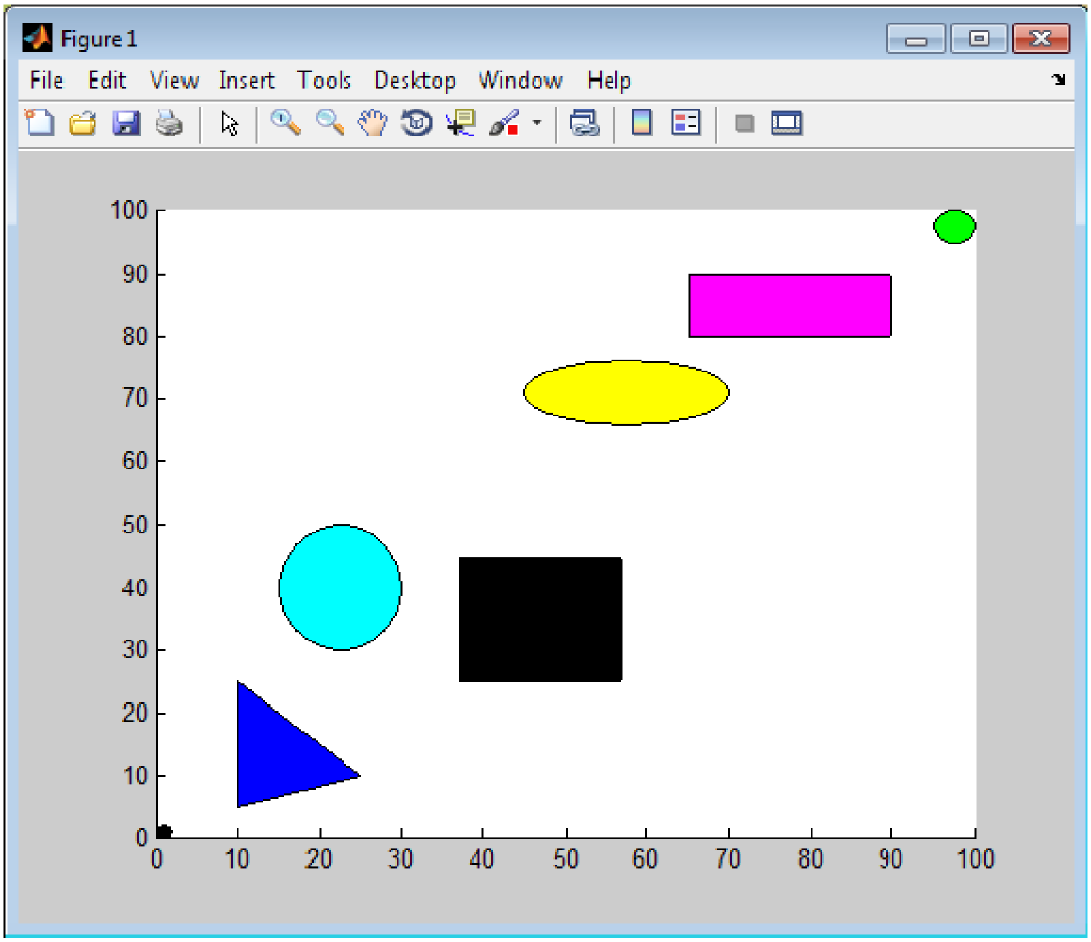


Fig 5.4 Figure showing the obstacles posed in the robots path

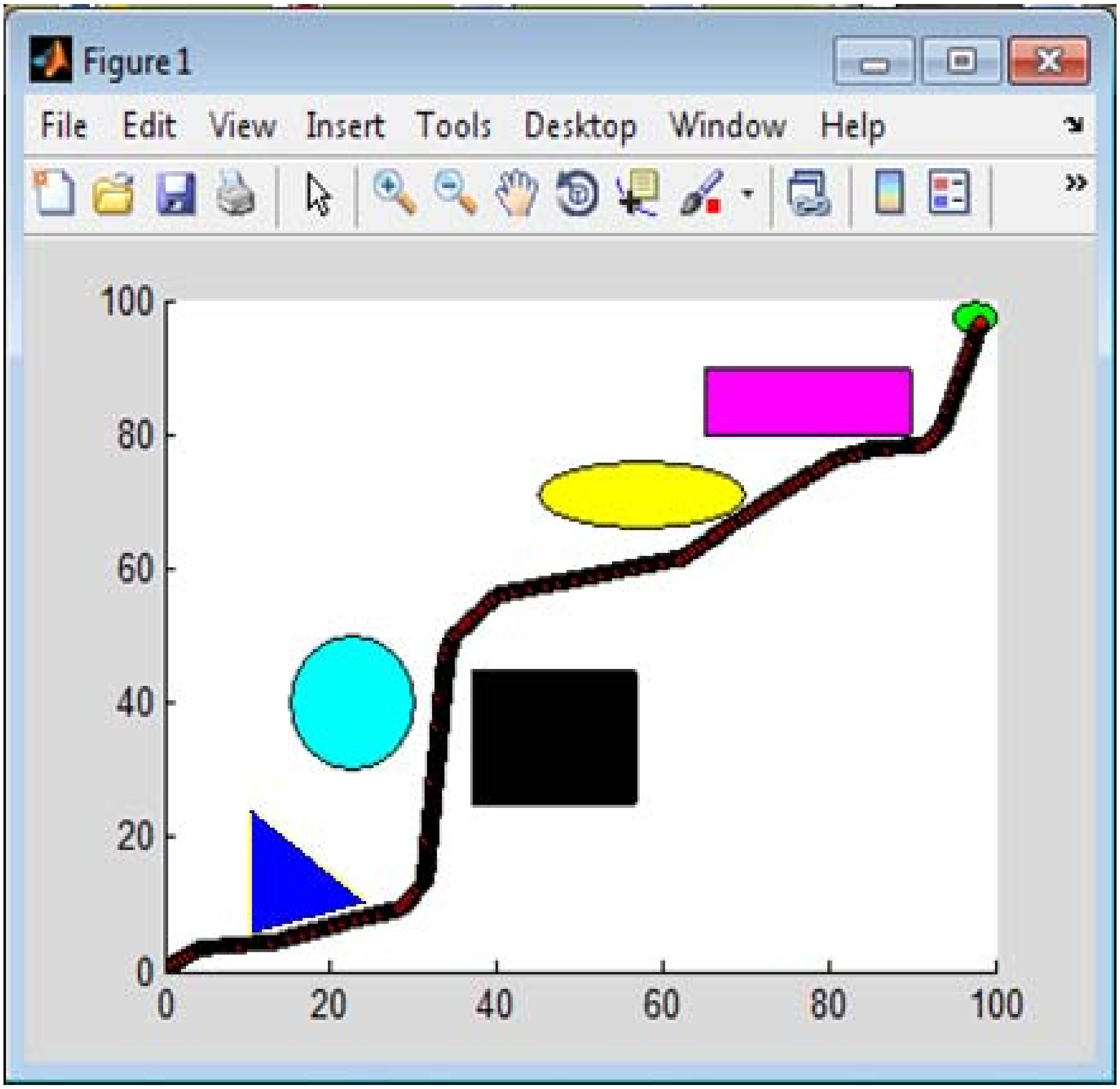


Fig. 5.5. Path taken by the robot during simulation in MATLAB for type 2 fuzzy

5.5 EXPERIMENTAL ANALYSIS

The path achieved by the robot during simulation was verified by implementing the program on a stingray robot in a physical environment similar to the environment produced in simulation. The path length of the robot was measured in the real environment as well as in the simulation.

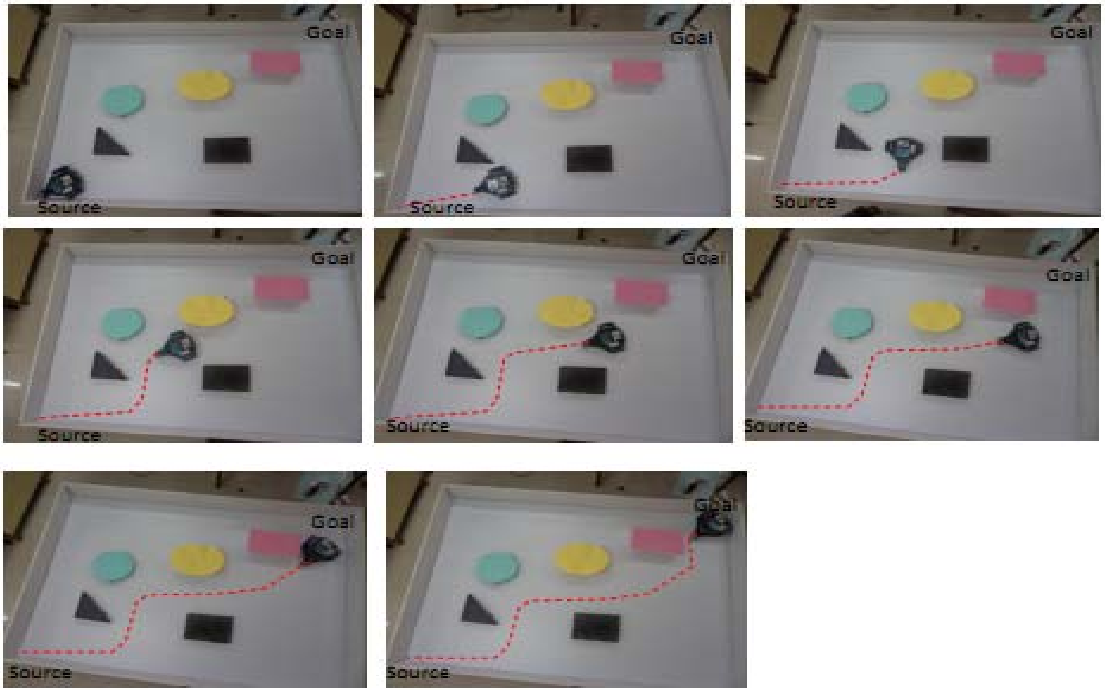


Fig. 5.6. Path taken by the robot during experiment using type 2 fuzzy controller

5.7 RESULTS AND DISCUSSION

The comparison between the simulation and experimental results was done in terms of the distance covered by the robot and the time taken. The comparison for Fuzzy logic is given in the following table:

Sl. No.	Algorithm used to find the robot's path	Path length in simulation achieved by proposed technique in cm.	Path length in experiment achieved by proposed technique in cm.	Error in %
1.	Type 2 Fuzzy logic controller	258.2	258.33	0.050

Table 5.1 Results found from the simulation and experiment for Type-2 Fuzzy Method

Scale for simulation: 1 = 27.777

5.8 CONCLUSION

The simulation of the above techniques in MATLAB revealed that the path taken by the robot in all situations is different. However the robot was successful in avoiding the obstacles in the path and was successful in reaching the goal position. The comparison between simulation and experimental results showed a good agreement.

CHAPTER 6

ARTIFICIAL BEE COLONY ALGORITHM FOR MOBILE ROBOT

6. ARTIFICIAL BEE COLONY FOR MOBILE ROBOT

Today's world has become completely mechanized and a greater part of it has already been occupied by machines. They perform almost each and every job that used to be done by humans prior to their invention. The most important machine that is also nearest to mankind is a robot. Their invention took place just like other machines but eventually scientists decided to give them some intelligence that would allow them to take their own decisions and think like other organisms. This quality or feature came to be termed as artificial intelligence. Now artificial intelligence is a widespread area of study since it has many aspects. A part of this area is the evolutionary computation and a subset of this is an evolutionary algorithm. It describes the process of evolution in organisms. The evolutionary algorithms use instances from nature for improving the intelligence of a robot. They focus on the mechanisms of navigation, target location and obstacle avoidance of various organisms found in nature. The present chapter focuses on the intelligence provided by one of such evolutionary algorithms, the bee colony algorithm.

6.1 INTRODUCTION

A bee colony consists of up to 60,000 to 80,000 bees in a single hive. The bees in the hive have one breeding Queen, some male drones, several thousands of sterile female workers and many young bee larvae or broods. Now the worker bees are divided into three main categories. The employed bees are bees that are engaged with a food source. The onlooker bees are those which are not appointed to any food sources yet. They stay in the hive and wait for the employed bees to show their dance so that they can choose the food source that they want to go to. The scout bees on the other hand are explorer bees that explore new food sources.

A colony of honey bees can extend itself over a very long distance (maximum up to 14 km) and simultaneously in multiple directions in order to exploit a larger number of food sources [149]. A colony can only prosper by employing its foragers in good fields. However, flower patches with abundance of nectar or pollen that can be collected with lesser effort would be visited by more bees rather than patches with less nectar or pollen would receive fewer bees [150]. The foraging process begins by scout bees or the worker bees in a colony being sent to search for new and untouched flower patches. There is randomness in their movement from one patch to another. During the harvesting season, a colony keeps a percentage of the population as scout bees so as to continue its exploration [151].

When the scout bees return to the hive, those bees that have found a patch rated above a certain quality threshold [152] (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform one of the two kinds of dance forms, round dance or the waggle dance [153]. Round dance is generally performed when there is food source nearby and the direction of the food source is given by this dance. Another form of dance is the waggle dance. This dance is essential for colony communication, and contains three pieces of information regarding a food source: the direction in which it will be found, its distance from the hive and its quality rating (or fitness). This information helps the colony to send its bees to flower patches precisely, without using guides or maps [179]. Each individual's knowledge of the outside environment is gathered from the waggle dance. This dance enables the colony to evaluate the importance of different patches according to both the quality of the food they provide and the amount of energy needed to harvest the required amount of food [180]. After waggle dancing inside the hive, the dancer (i.e. the scout bee) goes back to the flower patch or the food source with the follower bees gathered from the hive after dancing that were waiting inside the hive [181]. Other follower bees are sent to other promising food sources. This allows the colony to gather food more quickly and efficiently [182].

While harvesting from a food source, the bees monitor its food level. This is necessary in order to decide upon the next waggle dance when they return to the hive [183]. If the patch is still good enough as a food source, then it will be advertised in the waggle dance and more bees will be recruited to that source.

6.2 ABC ALGORITHM

Artificial Bee Colony Algorithm (ABC) is an optimization algorithm based on the intelligent foraging behaviour of honey bee swarm [184]. This model that leads to the emergence of collective intelligence of honeybee swarms consists of three essential components: food sources, employed foragers, and unemployed foragers, and defines two leading modes of the honeybee colony behaviour: recruitment to a food source and abandonment of a source [185]. Communication among bees related to the quality of food sources occurs in the dancing area. The related dance is called waggle dance [13].

The main steps of the algorithm are as below:

1. Initialize Population
2. Repeat
3. Place the employed bees on their food sources
4. Place the onlooker bees on the food sources depending on their nectar amounts
5. Send the scouts to the search area for discovering new food sources
6. Memorize the best food source found so far
7. Continue until requirements are met

In first step, the bee algorithm starts with the scout bees (n) being placed randomly in the search space. Step 2 continues the fitness of the sites visited by the scout bees and evaluates them. The employed bees are placed on their food sources in step 3. In step 4, bees that have the highest fitness are chosen as “selected bees” and sites visited by them are chosen for the neighbourhood search. Then, in steps 5 and 6, searches are conducted in the neighbourhood of the selected sites, thereby assigning more bees to search near to the best food sites. The bees are either chosen directly according to the fitness associated with the sites they are visiting or the fitness values are used to determine the probability of the bees being selected. Searches in the neighbourhood of the best food sites that represent more promising solutions are done in a more detailed way by recruiting more bees to follow them than the other selected bees. Apart from scouting, this differential recruitment is a key operation of the Bee Algorithm. However, in step 6, for each food source only the bee with the highest fitness will be selected to form the next bee population although in nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of every iteration the colony will have two parts to its new population – those that were the fittest representatives from a patch and those that have been sent out randomly. The whole process can be seen in the flowchart is given in figure1

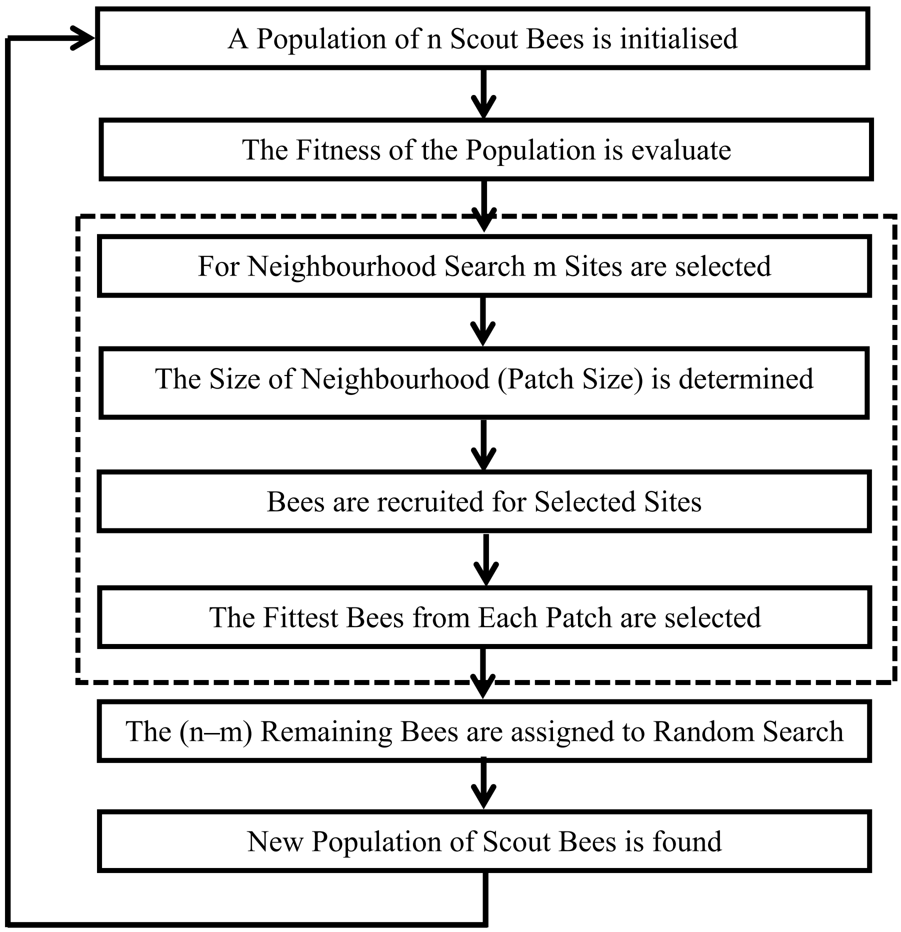


Fig.6.1. Flowchart showing the bee algorithm

As per the given problem, the values are set prior to the algorithm execution. The four inputs are taken as the food sources and the range of distribution is taken as (-5, 5). The fitness function taken is given by

$$fit_i = \begin{Bmatrix} \dfrac{1}{1+f_i} \ \ if\ f_i \geq 0 \\ 1+abs(f_i)\ if\ f_i \leq 0 \end{Bmatrix} \tag{6.1}$$

The maximum fitness value is calculated and the quality of the best food source is achieved. The cycle is repeated using the equation

$$v_{i,j} = x_{i,j} + \Phi_{ij}(x_{i,j} - x_{k,j})\ v_{i,j} = x_{i,j} + \Phi_{ij}(x_{i,j} - x_{k,j}) \tag{6.2}$$

Where k=1, j=0 and is taken randomly in between [-1,1].

Again the fitness value is calculated and the greediness value is calculated. If the greediness value is greater than the fitness value then the cycle is repeated for the next bee in the population. This process is carried out for all the onlooker bees and then then cycle value is increased.

6.3 SIMULATION

These values were given to the MATLAB simulator and a program for the navigation of the robot was created. The microchip in the robot was instilled with the program written in the MATLAB simulator. Various obstacles were provided in the path of the robot so that it had to change its path to reach its destination. The figure showing the arrangement of the obstacles is shown in the figure. The robot can be seen as a small black dot at the origin of the graph whereas the destination is seen as the green circle at the other end of the graph. The robot has to dodge the obstacles and reach the destination.

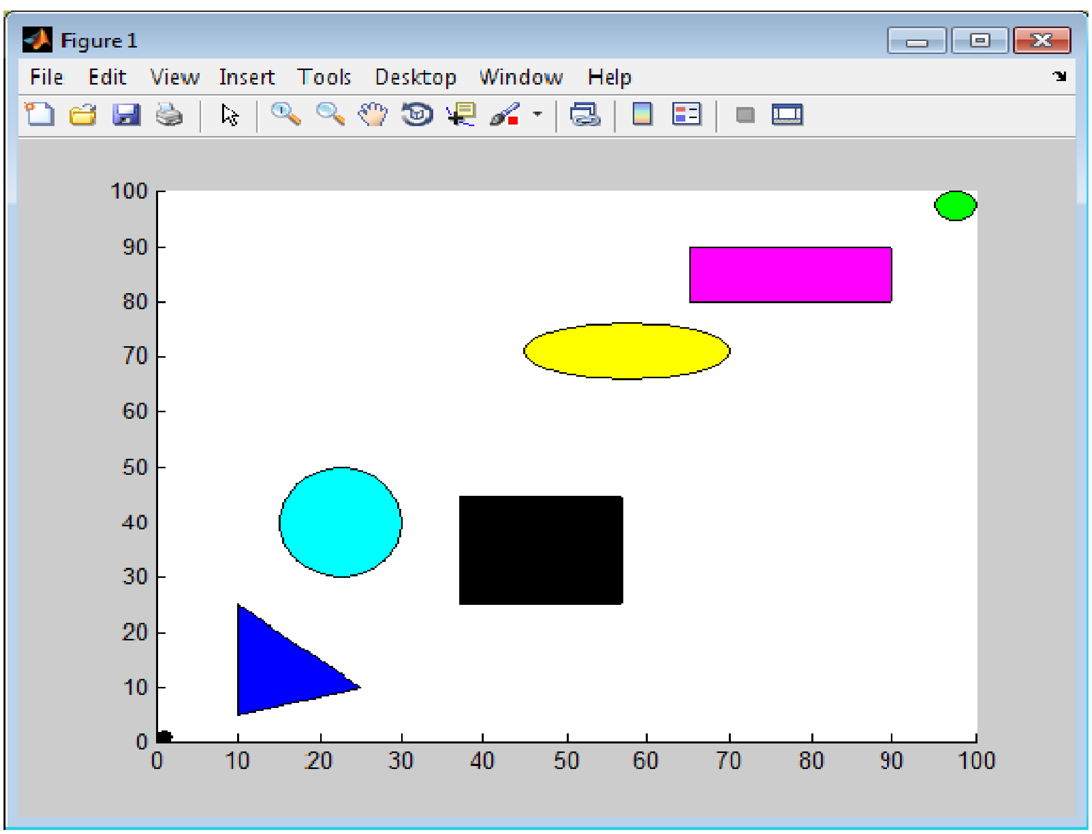


Fig. 6.2. Figure showing the arrangement of obstacles in the environment

The program allows the robot to reach the destination safely without colliding with the obstacles. Figure shows the path of the robot taken to reach the destination.

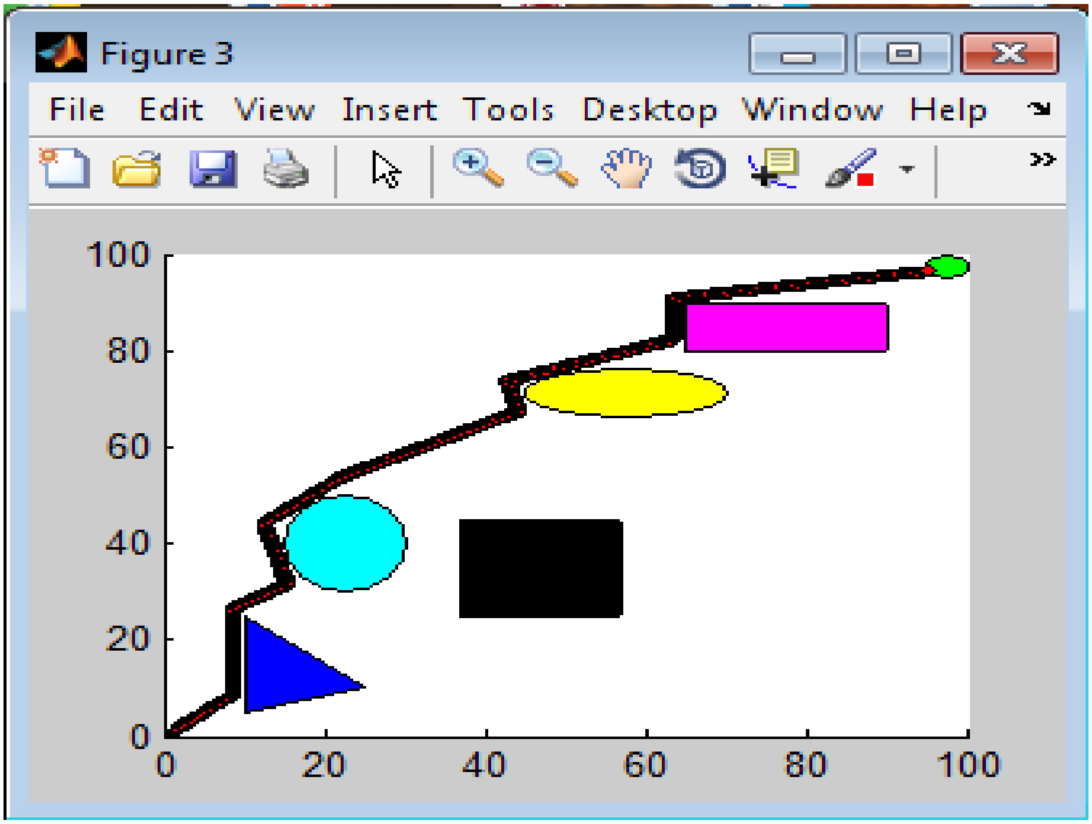


Fig.6.3. Path taken by the robot during simulation in MATLAB for ABC.

6.4 EXPERIMENTAL ANALYSIS

The path achieved by the robot during simulation was verified by implementing the program on a stingray robot in a physical environment similar to the environment produced in simulation. The path length of the robot was measured in the real environment as well as in the simulation.

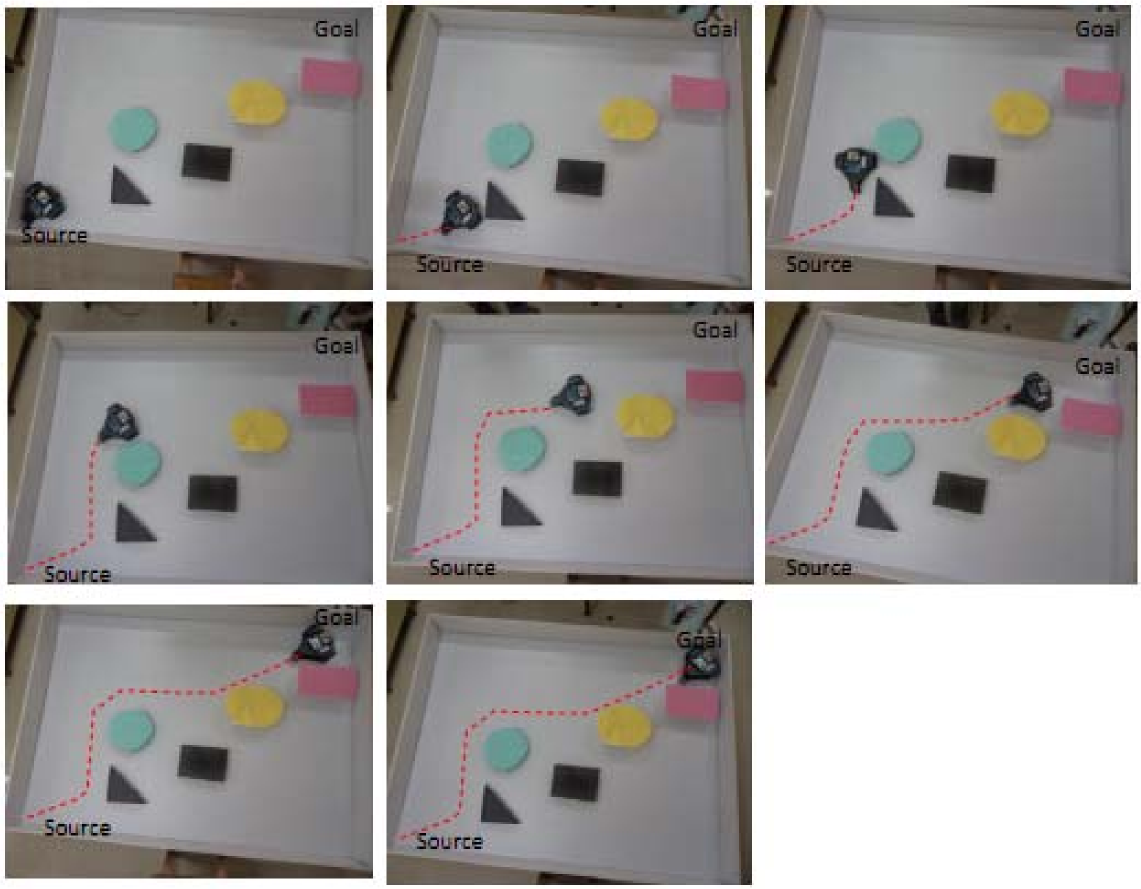


Fig.6.4. Path taken by the robot during experiment using ABC controller.

6.5 RESULTS AND DISCUSSION

The comparison between the simulation and experimental results was done in terms of the distance covered by the robot and the time taken. The comparison for Fuzzy logic is given in the following table:

Sl. No.	Algorithm used to find the robot's path	Path length in simulation achieved by proposed technique in cm.	Path length in experiment achieved by proposed technique in cm.	Error in %
1.	Artificial bee colony algorithm	249.3	250.0	0.280

Table 6.1. Results found from the simulation and experimental results for ABC method

Scale for simulation: 1 = 27.777

6.5 CONCLUSION

The simulation of the above techniques in MATLAB revealed that the path taken by the robot in all situations is different. However the robot was successful in avoiding the obstacles in the path and was successful in reaching the goal position. The comparison between simulation and experimental results showed a good agreement.

CHAPTER 7

NEURAL NETWORKS FOR MOBILE ROBOTS

7. NEURAL NETWORKS FOR MOBILE ROBOT

A basic research in the field of pattern recognition involves focus on network architectures. Neural network methods are considered as generalizations of classical pattern oriented methods in statistics and various areas of engineering. Artificial neural networks are inspired by the early models of sensory processing by the brain. A neural network is a graph, with patterns as numerical values attached to the nodes of the graph and transformations between patterns achieved by simple message-passing algorithms. This chapter provides a novel approach for design of an intelligent controller for autonomous mobile robot using multilayer feed forward neural network which enables the robot to navigate in a real world dynamic environment.

7.1 INTRODUCTION TO NEURAL NETWORKS

Artificial Neural Network (ANN) is a paradigm for information processing that is motivated by the way information is processed in a biological neural network, such as the brain. The central block of this paradigm is the distinctive structure of the information processing system. It consists of a large number of highly interconnected and organized processing elements (neurons) working together to process and solve. Like human beings, ANNs learn by lots of examples. Usually an ANN is implemented for a specific task, such as pattern approbation, data classification or obstacle avoidance through a learning process.

A novel thought was given for tracking control of the mobile robots by the Neurodynamics approach to get a smooth velocity of the robot with a good backward movement. For a nonholonomic mobile robot's real time navigation by incorporation of a back stepping technique, a Neurodynamics model has been designed. Contrasting to other tracking control techniques, this novel approach is better in generation of continuous smooth control of robot signals with zero initial velocity, for which the theory of Lyapunov stability has been used. Moreover this can also be used very smoothly in the situations with large tracking errors.

7.2 BASIC NEURON AND NETWORK LAYER

An artificial neuron is the component which receives a lot of inputs and provides single output. Two modes of maneuver are basically available; the training mode and the using mode. In the training mode, the firing of neuron can be trained for particular input patterns. In the using mode, based on the taught input, the taught output becomes the current one. If the input pattern has not been taught, the firing rule determines the decision of firing. The artificial neural networks consist of three layers: input layer, hidden layer and output layer.

1. The input units convert the raw information that is fed into the network.

2. Each hidden unit output is determined by input units and the connection weights between them.

3. The output units' behaviour depends on the hidden units' output and the weights between them. The hidden units are free to put up their own representations of the input. The activity of hidden units is determined by the weights, and so with modification of weights, a hidden unit can choose what it represents.

7.3 THE LEARNING PROCESS

The remembrance of patterns and the succeeding network response can be categorized into two general paradigms. In associative mapping the network learns to produce an output on the set of input units whenever another output is applied on the set of input units. The associative mapping is of two types.

- ❖ **Auto-association**: An input pattern is associated with itself and the pattern output units match the trained one. This is used for other pattern completion, i.e. to produce a pattern when some portion of it or a partial pattern is presented. In the second case, the network basically stores pairs of patterns associating two sets of patterns.

- ❖ **Hetero-association:** It is associated with two recall mechanisms: Nearest-neighbour recall, where the output produced corresponds to the stored input pattern, closest to the pattern presented interpolative recall, where the output pattern leads to interpolation which is based on similarity dependence of the patterns stored compared to the pattern presented. Another model, which is a variant associative mapping, is classification, i.e. when input patterns are to be classified into a fixed set of classifications. Every neural network owns knowledge which is contained in the weights. A learning rule for altering the values of the weights must lead to modification of the knowledge stored in the network as a function of practice. Data is stored in the weight matrix W of a neural network. The determination of the weights is called the learning process. Following the way learning is performed, we can categorize two major categories of neural networks:

- **Fixed networks:** The weights are fixed, i.e. the change of time is zero. In such networks, the weights are computed according to the problem to be solved.

- **Adaptivenetworks:** Here the weights are changeable, i.e. the change of weights with respect to time is non- zero. Entirely, the learning methods used for adaptive neural networks can be classified into two chief categories:

  - **Supervised learning:** which includes an external teacher, so that each output unitis told its expected response to input signal. In the learning process global information may be a requirement. Models of supervised learning include error-correction learning, reinforcement learning and stochastic learning. A critical issue about supervised learning is the problem of error convergence, i.e. the minimization of difference between the desired and computed values. The objective is to find out a set of weights that reduces the error, for which some methods is common to many learning methods, is the least mean square (LMS) convergence.
  - **Unsupervised learning:** uses no external teacher and is centered upon local information. It is also stated to as self-organization, in the way that it self-organizes data presented to the network and detects their emergent group properties. Examples of unsupervised learning are Hibbing learning and competitive learning.

7.4 TYPES OF NEURAL NETWORK

1. **Feed-forward neural network:** The first and probably the simplest type of artificial neural network invented, this network features the movement of information in single direction. From the input nodes information goes to the hidden nodes (if any) and finally to the output nodes. There are no cycles or loops in the network. Feed-forward networks can be constructed from various types of units, e.g. binary McCulloch-Pitts neurons, one example being the perception.

2. **Learning Vector Quantization:** Learning Vector Quantization (LVQ) may be understood as neural network architecture. Representatives of the classes parameterize in LVQ, together with a suitable distance measure, a distance-based classification scheme.

3. **Recurrent neural network**: Contrary to feed-forward networks, recurrent neural networks (RNNs) are models with bi-directional data flow. While there is linear transmittance of data in feed-forward network from input to output, RNNs also propagate data from later units to earlier units. RNN scan be used as general sequence processors.

4. **Fully recurrent network:** It is the basic architecture established way back in the 1980s: consisting of a network of neuron-like units, each with a directed link to every other unit, with all units having a time-varying real-valued initiation. Each assembly has a modifiable real-valued weight. Some of the nodes are called input nodes; some are called output nodes, the rest hidden nodes. Most architecture below is special cases.

5. **Hopfield network:** The Hopfield network is of historic interest although it is not all-purpose RNN, as it is not designed to process systems of patterns. Instead it requires static inputs. It is an RNN in which all contacts are symmetric. Designed by John Hopfield in 1982, assures that its dynamics will converge. If the links are trained using Hebbian learning then the Hopfield network will perform as robust content-addressable memory, resistant to connection alteration.

6. **Simple recurrent networks:** This special case of the Hopfield network was when a three-layer network is used, along a set of ”context units” in the input layer. There are links from the hidden layer or from the output layer to the context units fixed with a weight of one. At each time step, the input is transmitted in a standard feed-forward fashion, and then a simple back prop-like learning rule is applied. The fixed back connections result in the context units always preserving a copy of the previous values of the hidden units (since they propagate over the connections before the learning rule is applied).

7. **Echo state network:** The echo state network (ESN) is a recurrent neural network with a sporadically linked random hidden layer. Only the weights of output neurons can be changed and be trained. ESNs are good at replicating certain time series.

7.5 THE BACK-PROPAGATION ALGORITHM

For training a neural network to execute any task, weights of each unit must be changed in such a manner that the difference between the desired output and the actual output is reduced. It requires that the neural network computes the error offshoot of the weights (EW). In other words, it must calculate the fluctuation of error occurring with a slight increase or decrease in each weight. The back propagation algorithm is a commonly used method for finding the EW.

The back-propagation algorithm is easiest to comprehend if all the units in the network are linear. The algorithm computes each EW by first determining the activation, the rate at which the error varies as the movement level of a unit is altered. For output units, the activation error is just the difference between the real and the desired output. To calculate this for a hidden unit in the layer just before the output layer, firstly all the weights amongst that hidden unit and output units to which it is linked are recognized.

Activation errors of those output units are computed and the products are added. This sum equals the activation error in the hidden layer penultimate to output layer. We can similarly compute the activation for other layers, effecting from layer to layer in the reverse direction of the propagational direction of activities through the network. Hence it is called back propagation. Once the activation error has been computed for a unit, it is simple to calculate the EW for each incoming link of the unit. The EW is the creation of the EA and the activity through the incoming link.

7.6 NEURAL NETWORK CONTROLLER

A new epitome of intelligent navigation systems must be enriched with some common features like: criteria for optimal performance and ways to optimize design, structure and control of robot. The growing need for the deployment of intelligent, highly autonomous systems has made it beneficial to carry out research on artificial neural networks due to its high learning capabilities with a high level of knowledge interpretability. Neural network is able to build comprehensive knowledge bases considering sensor-rich system with real time constraints by adaptive learning, rule extraction and inclusion. The training for back propagation algorithm and its navigational performances analysis has been done in real experimental setup. As experimental result matches well with the simulation result, the realism of method is verified. To reduce travel time as well as the distance travelled, four layer perception neural networks has been designed by using the environmental information to make navigational decisions.

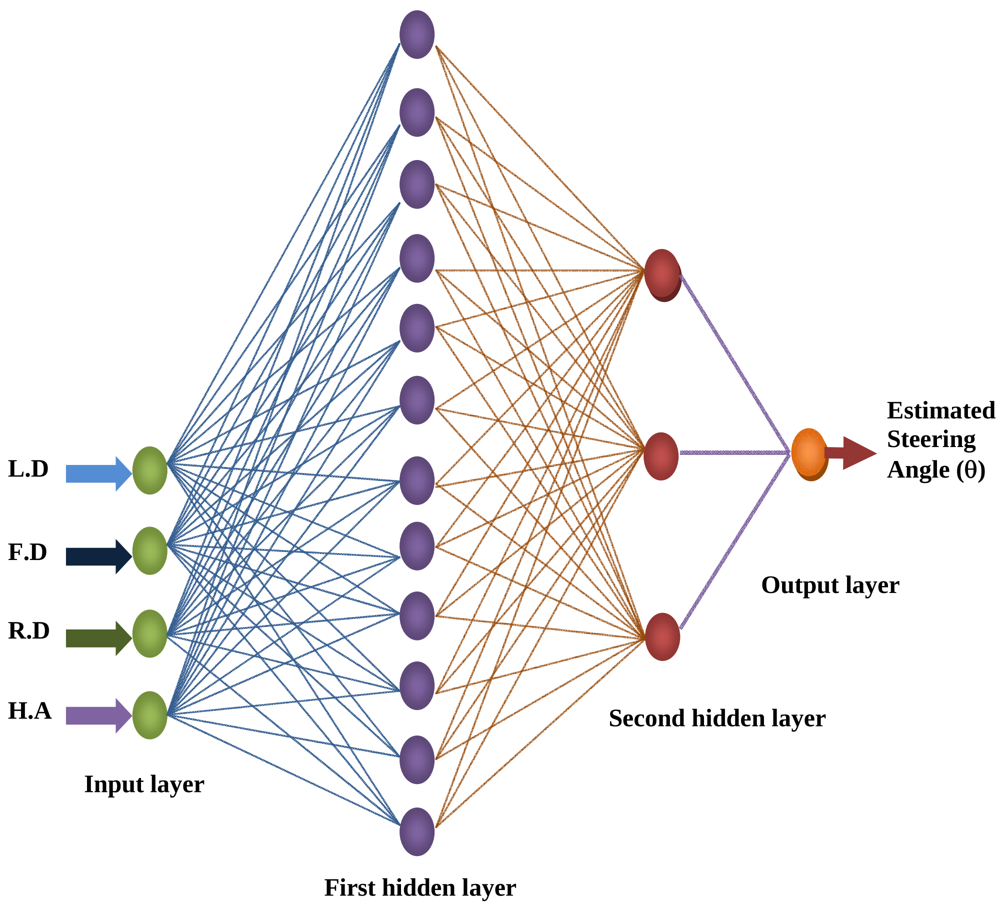


Fig 7.1. Multilayer Neural Controller for implementation of robotic behaviours

The first layer is used as input layer which has three neurons, for receiving the values of the distances from obstacles in front, left and right of the robot. Next the robot network consists of two hidden layers as shown in figure which adjusted the weight of neuron; as with one hidden layer it is difficult to train the network within a specified error limit. The training error is the difference between desired output and actual output. The first hidden layer has eighteen neurons and the second hidden layer has five neurons as shown in figure. These numbers of hidden layers were also found empirically. Then a output layer with a single neuron which provide steering angle to control the direction of movement of the robot. Back propagation method is used to minimize the error and optimize the path and time of mobile robot to reach the target.

During training and during normal operation, the input patterns fed to the neural network comprise the following components:

$$Y_1^{\{1\}} = \textit{Left obstacle distance from the robot} \quad (7.11)$$

$$Y_2^{\{1\}} = \textit{Right obstacle distance from the robot} \quad (7.12)$$

$$Y_3^{\{1\}} = \textit{Front obstacle distance from the robot} \quad (7.13)$$

$$Y_4^{\{1\}} = \textit{Target bearing} \quad (7.14)$$

These input values are distributed to the hidden neurons which generate outputs given by:

$$Y_j^{\{lay\}} = f\left(V_j^{\{lay\}}\right) \quad (7.15)$$

$$where, V_j^{\{lay\}} = \sum_i W_{ji}^{\{lay\}} Y_i^{\{lay-1\}} \quad (7.16)$$

lay : layer number (2 or 3)

j : label for j^{th} neuron in hidden layer 'lay'

i : label for i^{th} neuron in hidden layer 'lay − 1'

$W_{ji}^{\{lay\}}$: Weight of the connection from neuron i in layer 'lay-1' to neuron j in layer 'lay'

$f(.)$: Activation function, chosen in this work as the continuous log-sigmoid function:

$$f(x) = \frac{1}{1 + e^{-\beta x}} \quad (7.17)$$

The sigmoid has the property of being similar to the step function, but with the addition of a region of uncertainty. Sigmoid functions in this respect are very similar to the input-output relationships of biological neurons, although not exactly the same. Figure given below is the graph of a sigmoid function. During training, the network output _actual may differ from the desired output $\theta_{desired}$ as specified in the training pattern presented to the network. A measure of the performance of the network is the instantaneous sum-squared difference between $\theta_{desired}$ and θ_{actual} for the set of presented training patterns:

$$Err = \sum_{\text{all training patterns}} (\theta_{desired} - \theta_{actual})^2 \tag{7.18}$$

The error back propagation method is employed to train the network. This method requires the computation of local error gradients in order to determine appropriate weight corrections to reduce error for the output layer, the error gradient is:

$$\delta^{\{4\}} = f'\left(V_1^{\{4\}}\right)(\theta_{desired} - \theta_{actual})^2 \tag{7.19}$$

The local gradient for neurons in hidden layer [lay] is given by

$$\delta_j^{\{lay\}} = f'\left(V_1^{\{lay\}}\right)\left(\sum_k W_{kj}^{\{lay+1\}} \delta_k^{\{lay+1\}}\right) \tag{7.20}$$

7.7 SIMULATION:

The robot was given a set of instructions to follow and the program was written in MATLAB simulator. The microchip in the robot was instilled with the program written in the MATLAB simulator. Various obstacles were provided in the path of the robot so that it had to change its path to reach its destination. The figure showing the arrangement of the obstacles is shown in the figure 7.2. The robot can be seen as a small black dot at the origin of the graph whereas the destination is seen as the green circle at the other end of the graph. The robot has to dodge the obstacles and reach the destination.

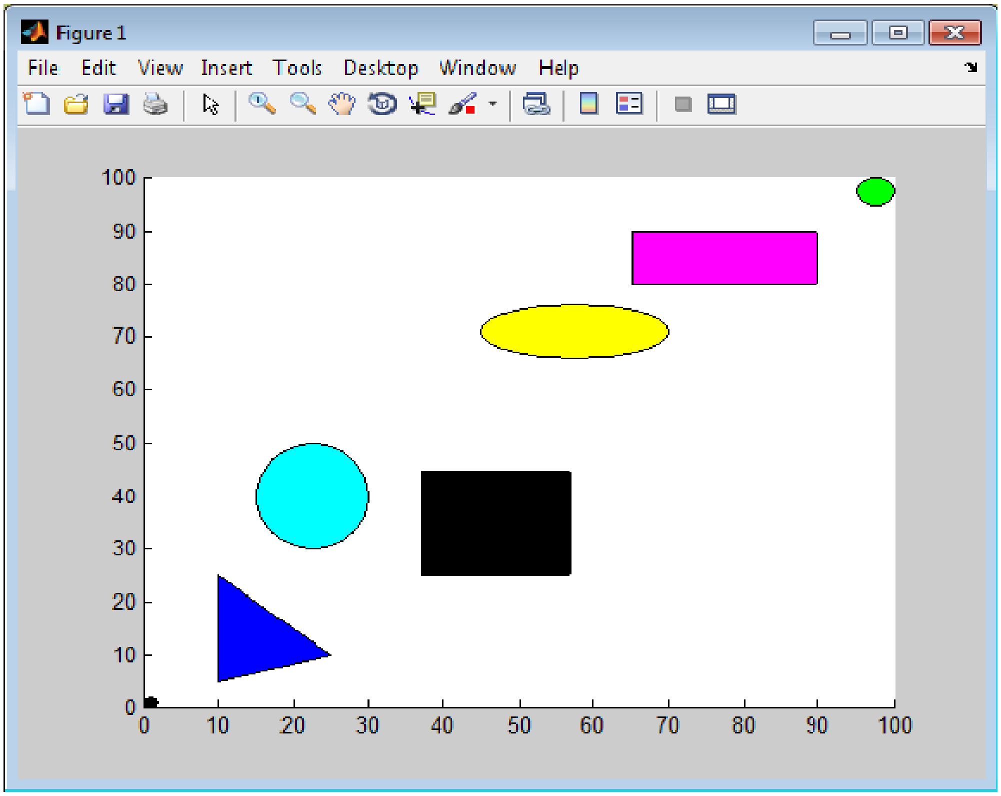


Fig. 7.2. Figure showing the arrangement of obstacles in the environment

The program allows the robot to reach the destination safely without colliding with the obstacles. Figure 7.3 shows the path of the robot taken to reach the destination.

7.8 EXPERIMENTAL ANALYSIS

The path achieved by the robot during simulation was verified by implementing the prograon a stingray robot in a physical environment similar to the environment produced in simulation. The path length of the robot was measured in the real environment as well as in the simulation.

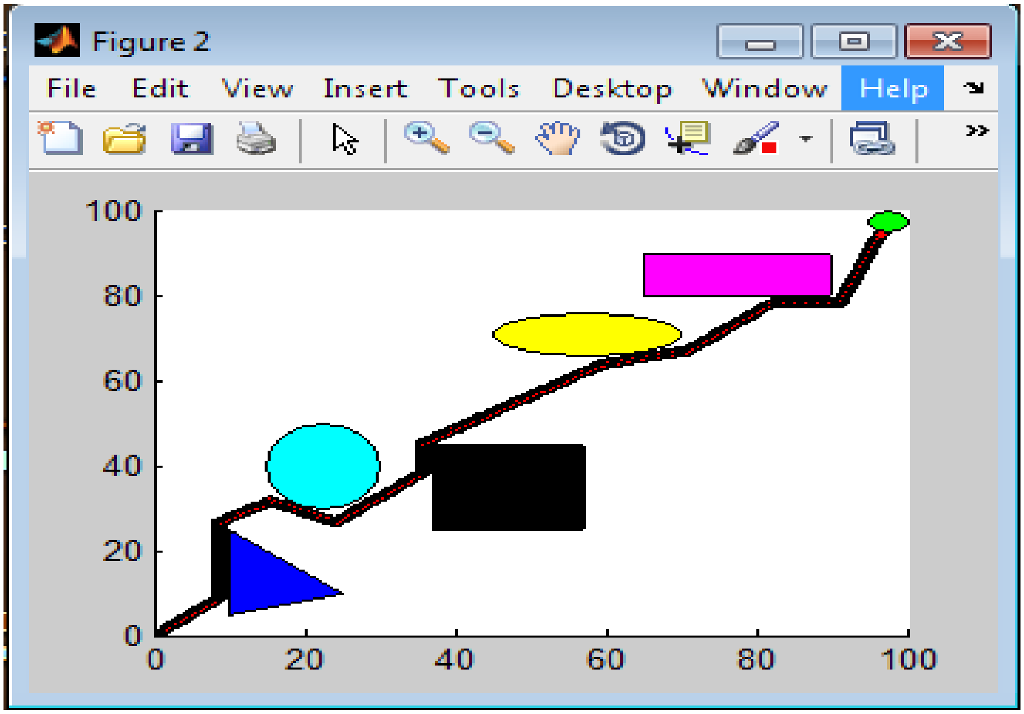


Fig.7.2. Path taken by the robot during simulation in MATLAB for NN

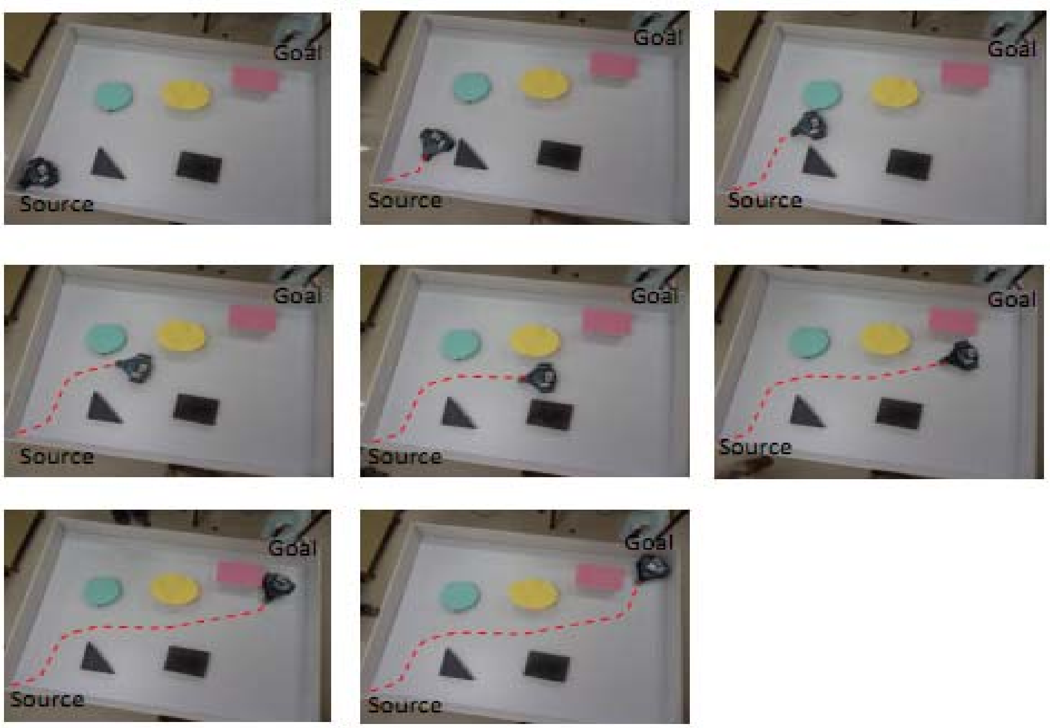


Fig.7.3. Path taken by the robot during experiment using NN controller.

7.9 RESULTS AND DISCUSSION

The comparison between the simulation and experimental results was done in terms of the distance covered by the robot and the time taken. The comparison for Fuzzy logic is given in the following table:

Sl.No.	Back propagation algorithm	Path length in simulation achieved by proposed technique in cm.	Path length in experiment achieved by proposed technique in cm.	Error in %
1.	Backpropagation algorithm	252.0	253.0	0.397

Table 7.1. Results found from the simulation and experimental results for NN method

Note:-Scale for simulation: 1 = 27.777

7.10 CONCLUSION

The simulation of the above techniques in MATLAB revealed that the path taken by the robot in all situations is different. However the robot was successful in avoiding the obstacles in the path and was successful in reaching the goal position. The comparison between simulation and experimental results showed a good agreement.

CHAPTER 8

HARDWARE ANALYSIS

8. HARDWARE ANALYSIS

The chassis is constructed from 1.58 mm thick 5052 Aluminum sheet metal. This robot has the specialty that there are three ultrasonic sensors having a wide field of view which can detect the obstacles and give a record to the micro-controller and the program will run accordingly. These sensors can detect the obstacles quickly in running condition also so that the robot can avoid the obstacles in front of it and move to the goal in an optimized path. The motors speed is 310 RPM at 7.2 V Dc which equates to approximately 2 m/s.

8.1 SPECIFICATIONS OF THE ROBOT

- Power requirements: Motors rated @ 7.2V (stall current 4.5A @ 7.2V)
- Motor details: 7.2VDC, 310 RPM, 6mm Shaft
- Operating temperature: 32 to 158 F (0 to 70 C)
- Dimensions, assembled: 13 Length 10.9 Width 5.5 Height (33 27.7 14 cm)
- Power requirements: 60mA @ 5 VDC (20mA each PING) sensor
- Communication: Positive TTL Pulse

8.2 FEATURES

- High-quality, scratch-resistant, type-2 anodized aluminum alloy chassis
- Solid 12.4 cm diameter wheels with durable high-traction rubber tread
- Durable 7.2 VDC motors have all metal gears. Two-wheel differential drive system with rear omni-directional wheel.
- Multiple mounting locations for sensors, add-ons, etc.
- 3 PING) Ultrasonic Range Finders
- Ultrasonic vision for the Stingray Robot for up to 9 meter.

These are some pictures of the Stingray robots:

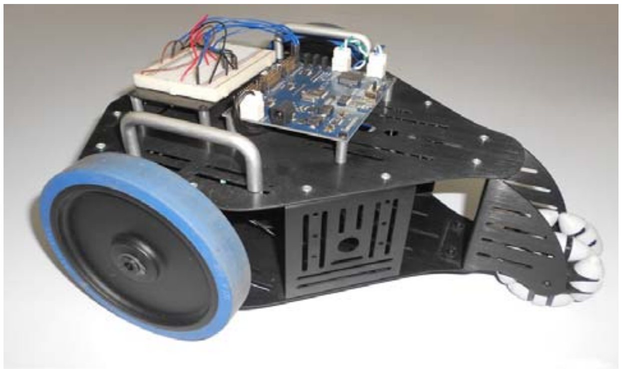


Fig.8.1. Side view of Stingray

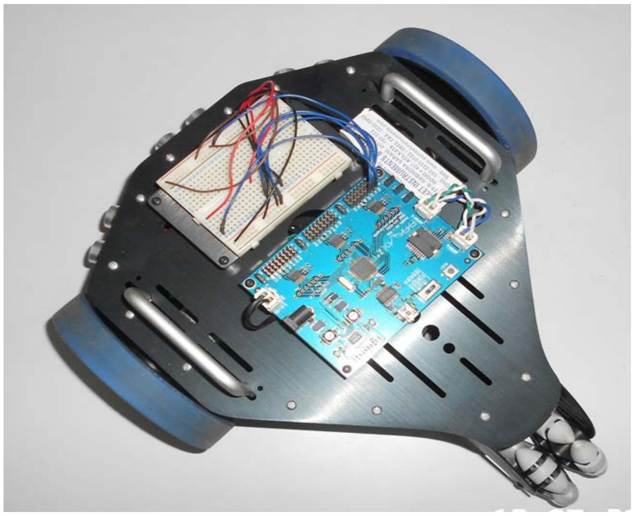


Fig.8.2. Top view of Stingray

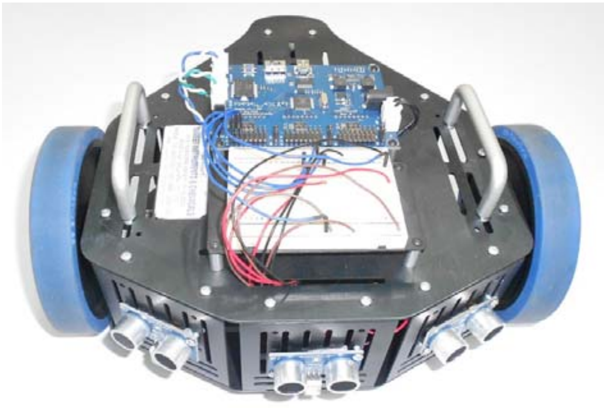


Fig.8.3. Front view of Stingray

CHAPTER 9

RESULTS AND DISCUSSIONS

9. RESULTS AND DISCUSSIONS

In the present findings a problem related to navigational path analysis of mobile robot in various environments has been analysed. Considering the kinematical aspect, AI techniques (e.g. Fuzzy Logic, Neural Network) are used for dynamic, optimized and collision-free path so that mobile robot can reach the target. This chapter, which is supposed to encapsulate the performances of present work done by mentioning the analysis and results of respective chapters for endorsement, can be divided into two main parts as following:

To grasp the expertise in performance, the self-adaptive robot's navigation and path planning algorithm must be consistent with the kinematics of the mobile robot. As sensory statistics is ineffective to provide vehicle‘s configuration, it becomes necessary to obtain stable kinematic and dynamic model for the robot in its global and local reference frame respectively.

In chapter three, Kinematic analysis of the mechanical structure of a robot has been done concerning the description of the motion with respect to a fixed reference cartesian frame by ignoring the forces and moments that cause motion of the structure. Modelling of mobile robot is done by combining all kinematic constraints for individual wheels. The different levels of designing wheeled mobile robot can be portrayed as: positioning of the robot model in the environment, maneuverability analysis and holonomicity checking with respect to kinematic constraints and generalized control of developed kinematic model. The manoeuvrability or degree of freedom deals with the possible motions that the robot may follow to reach a final configuration. Modelling of mobile robot with differential drive wheels as control systems has been addressed with a differential geometric point of view by considering only the conventional theory of "rolling without slipping". Such a robot can rotate on the spot (i.e., without moving the midpoint between the wheels), only if the angular velocities of the two wheels are equal and opposite.

Navigational analysis of mobile agent faces many critical troubles in real world environment than the problems regarding kinematic instability of the robot configuration. Selection of navigational techniques is very much significant in the research area of mobile robotics. All forms of robotic behaviours depend on intelligence of the controller to get collision free navigational path.

Fuzzy navigation technique allows a reactive control to move in a reasonable direction and velocity maneuvers of the autonomous mobile robot, is activated to achieve reasonable behavioural performance in static terrains. A Mamdani based Fuzzy logic controller (FLC has been recognised to be more compatible with the reasoning process of human behaviours. Fuzzy behaviour-based architecture for mobile robot navigation in unknown environments incorporates design of rule base considering basic behaviours for mobile robot navigation, i.e., goal seeking, obstacle avoidance, wall following and deadlock disarming etc.

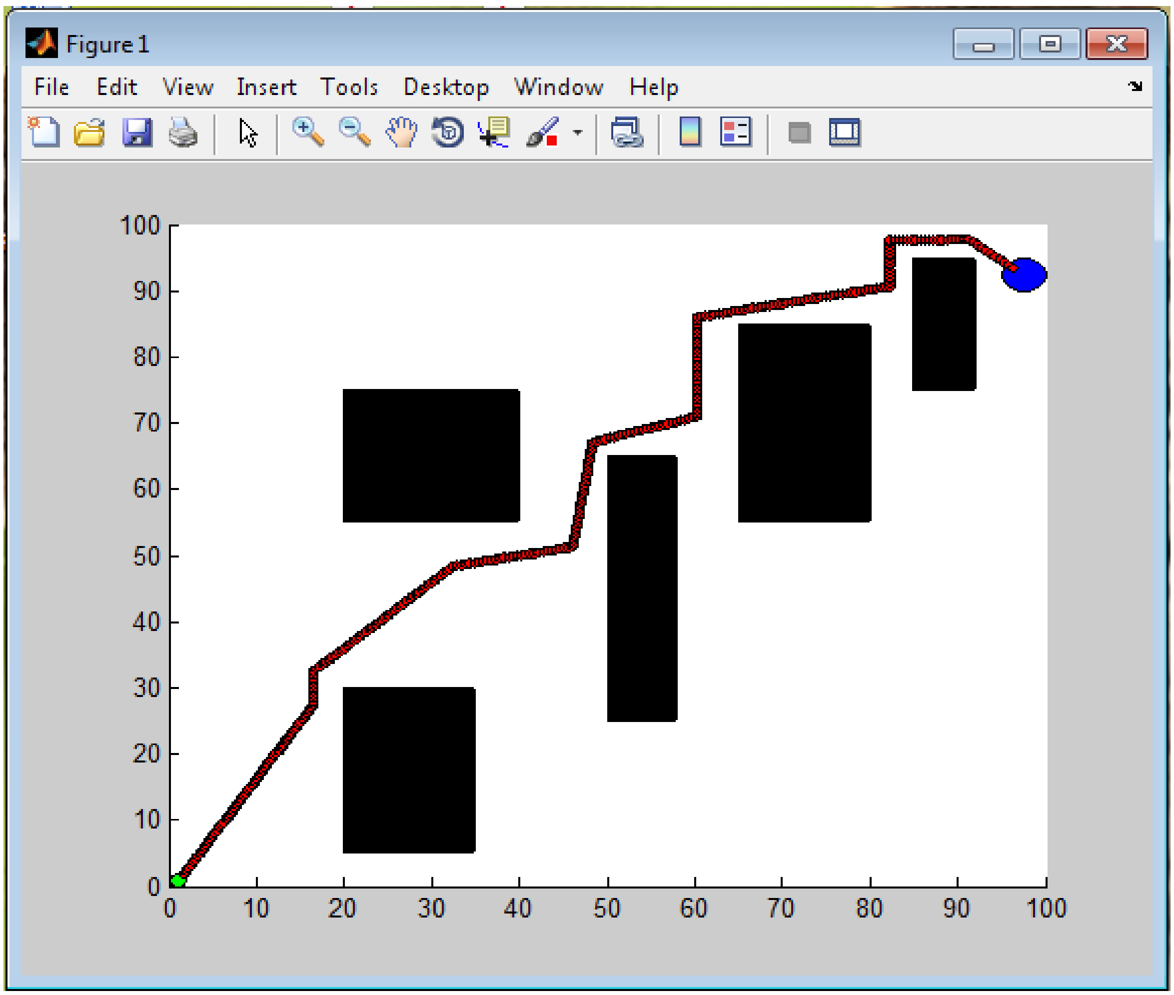


Fig. 9.1. Path taken by the robot during simulation in MATLAB for Fuzzy logic.

The movement of the mobile robot in another environment can also be viewed so as to confirm the results given in the chapter. The figure 9.1 shows the navigation of the mobile robot in another environment and in simulation as well as in experimental analysis using the fuzzy logic controller

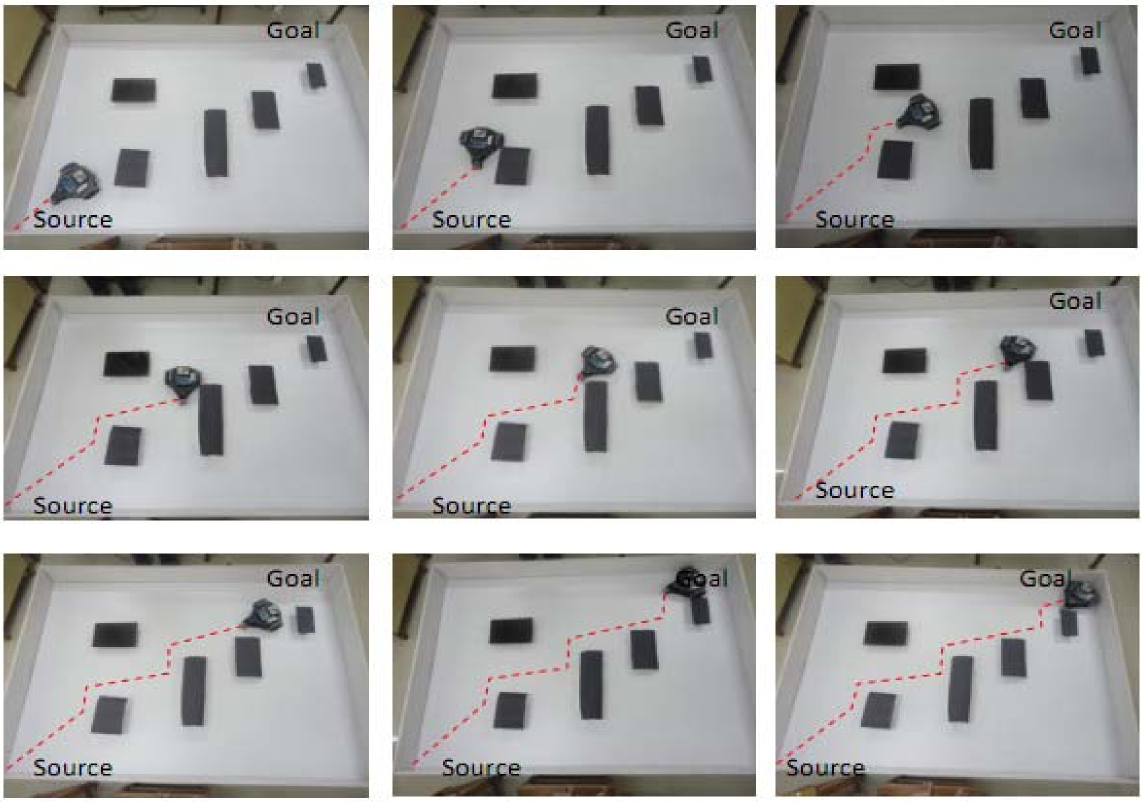


Fig. 9.2. Path taken by the robot during experiment using Fuzzy controller.

The next chapter gives an advanced version of the fuzzy logic behaviour called the type 2 fuzzy behaviour which enables us to remove more uncertainties from the decision making process. The results obtained from the experiments showed the above statement to be true. The truth can also be confirmed by looking at the navigation of the robot in another environment using the same type 2 fuzzy mechanism. The figure 9.2 shows the navigation of the mobile robot in another environment and in simulation as well as in experimental analysis using the type 2 fuzzy logic controller.

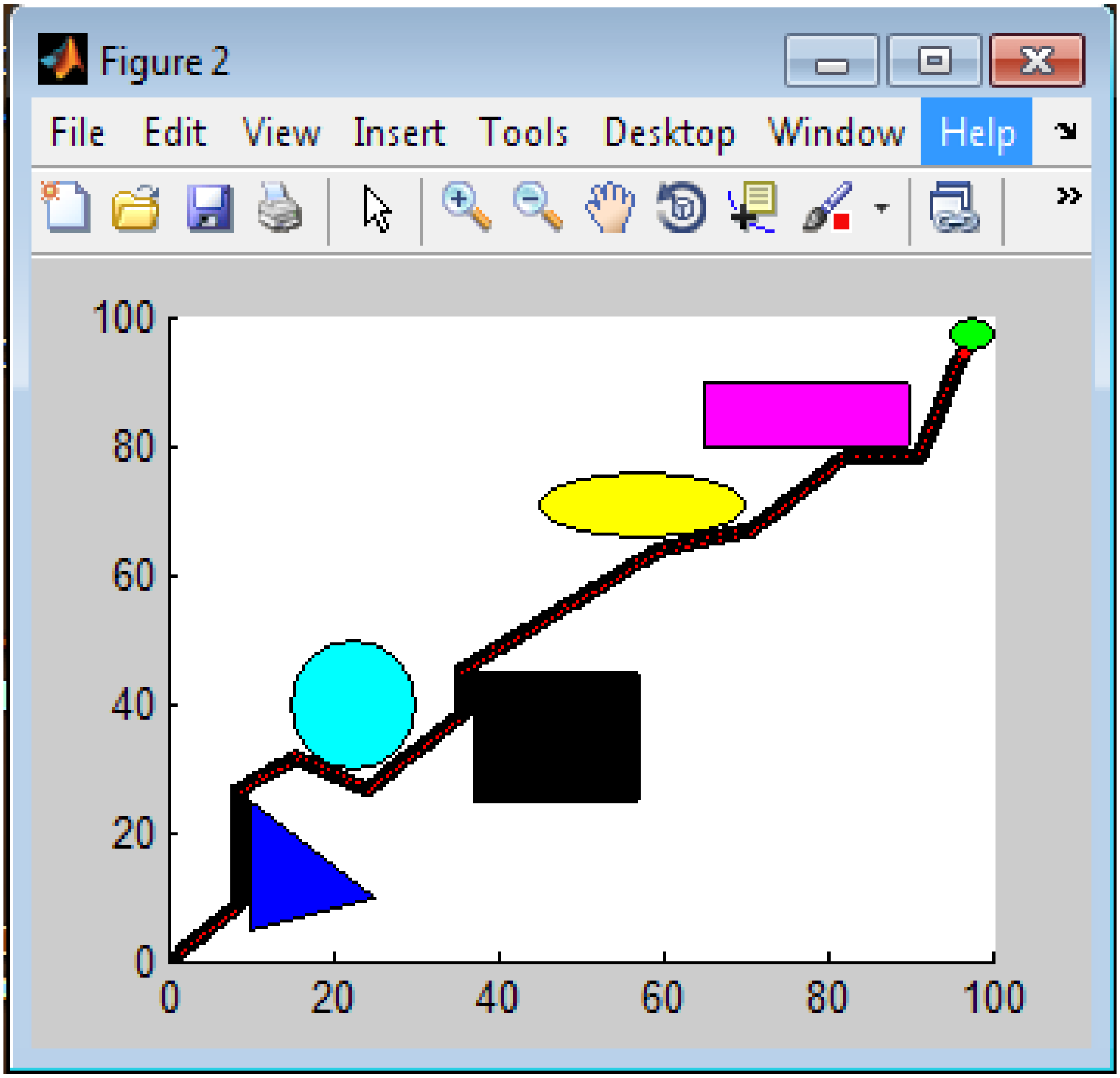


Fig. 9.3. Path taken by the robot during simulation in MATLAB for type 2 fuzzy

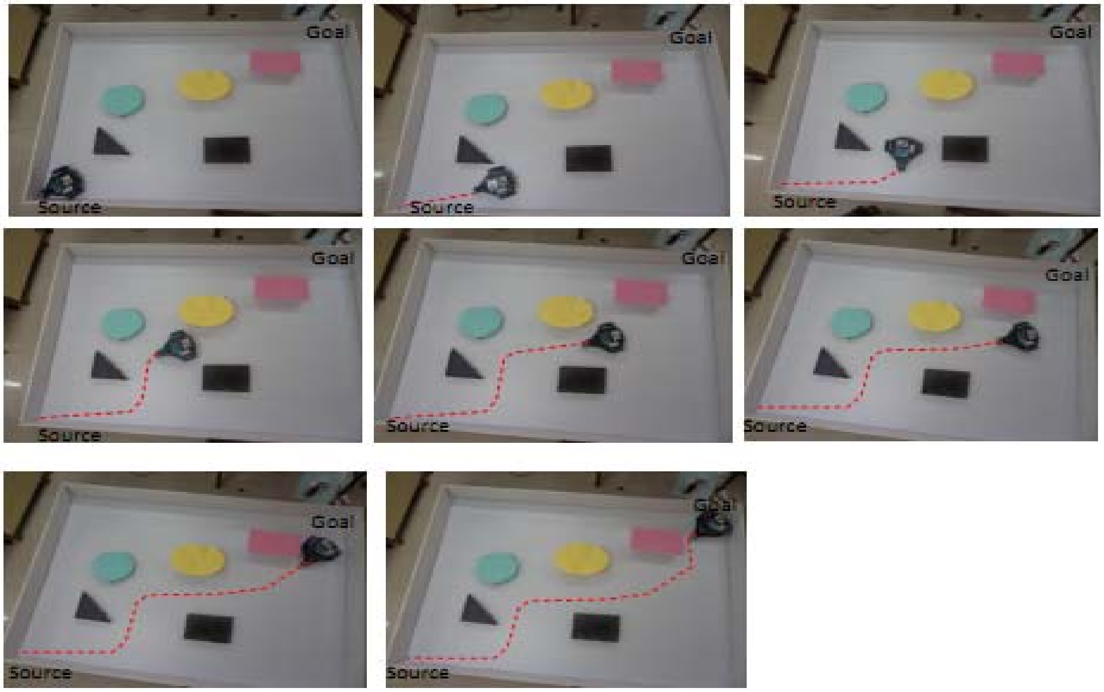


Fig. 9.4. Path taken by the robot during experiment using Fuzzy Type-2 controller.

In the next chapter, an evolutionary algorithm, the artificial bee colony algorithm has been used to make the robot move from the source to the goal. The robot undergoes a number of cycles to reach an optimised path. The path taken is shorter and more sensitive to the environment. The results can be seen in the figure given below. The figure 9.3 shows the navigation of the mobile robot in another environment and in simulation as well as in experimental analysis using the artificial bee colony algorithm.

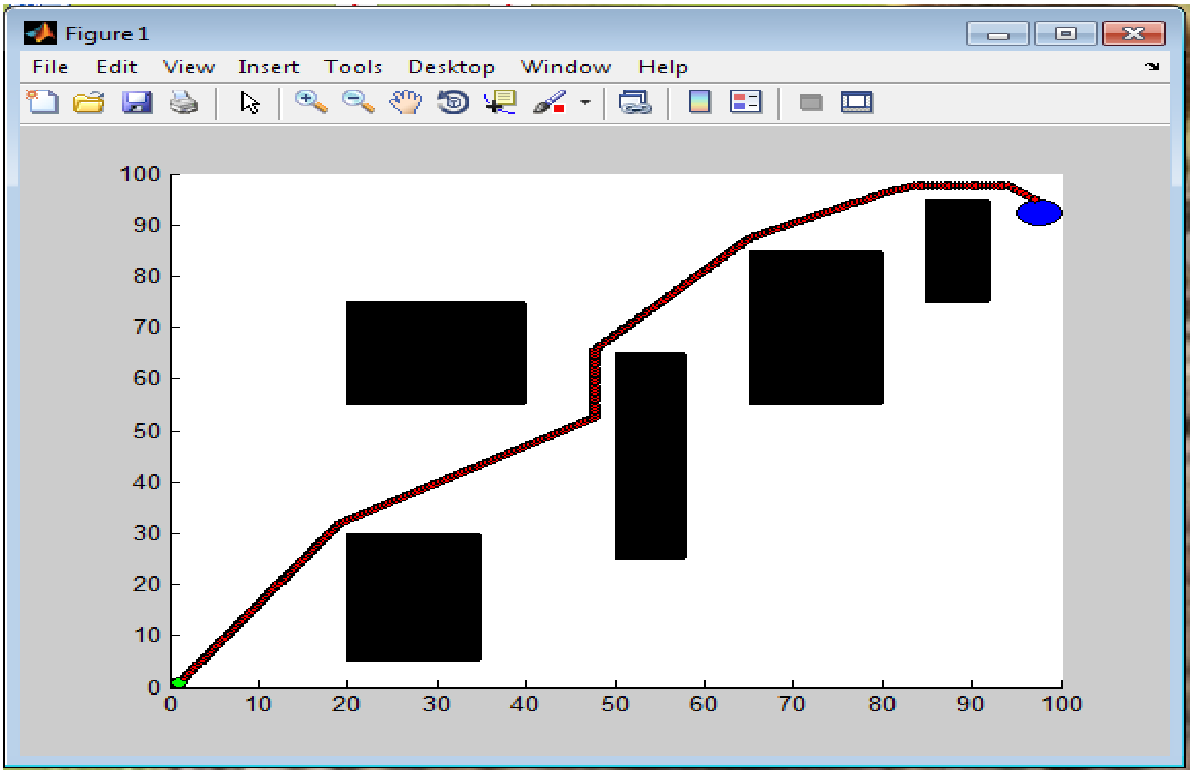


Fig. 9.5. Path taken by the robot during simulation in MATLAB for ABC.

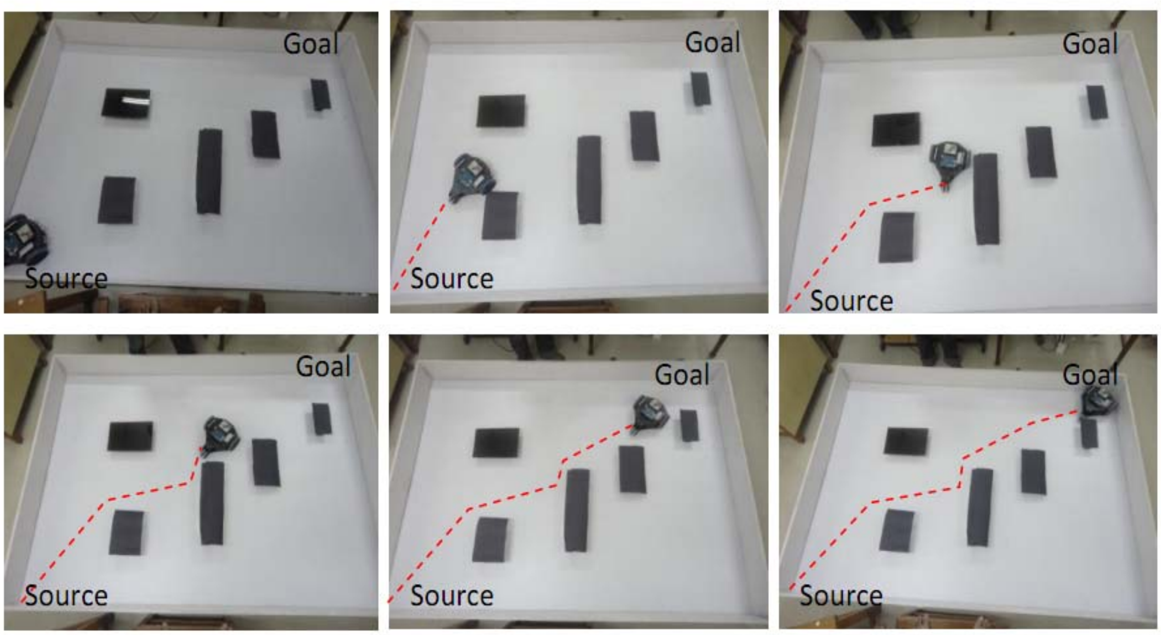


Fig. 9.6. Path taken by the robot during experiment using ABC controller.

A multilayer feed forward neural network using the principle of back propagation algorithm has been employed in the next chapter to increase the accuracy in steering angle measurement by eliminating uncertainties in target sensor reading. 200 training are used for designing an intelligent well trained neural controller for mobile robot being used to navigate in a cluttered environment. The results show the improvement in the results obtained from earlier mechanisms. The figure 9.4 shows the navigation of the mobile robot in another environment and in simulation as well as in experimental analysis using the neural network controller.

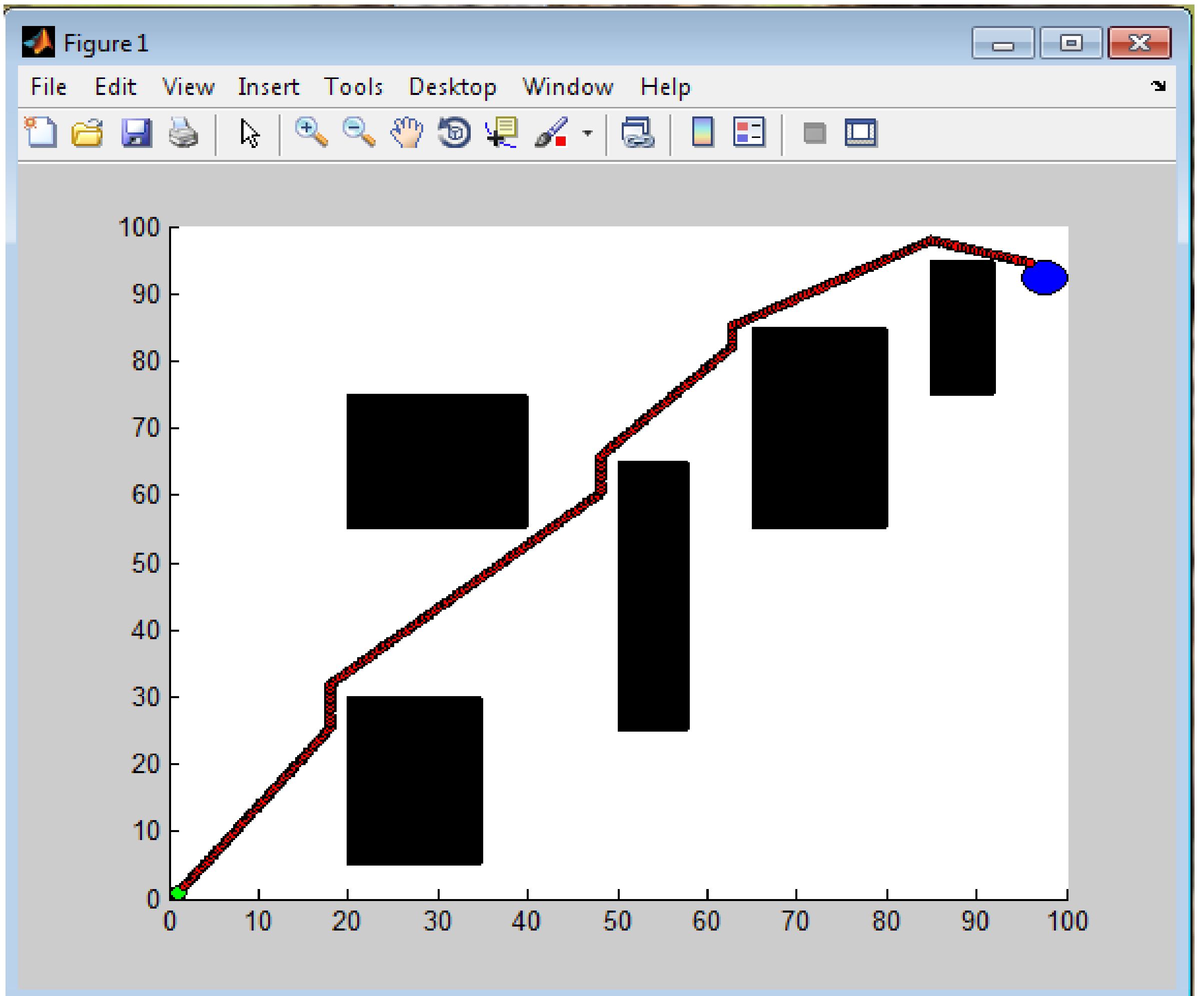


Fig. 9.7. Path taken by the robot during simulation in MATLAB for NN.

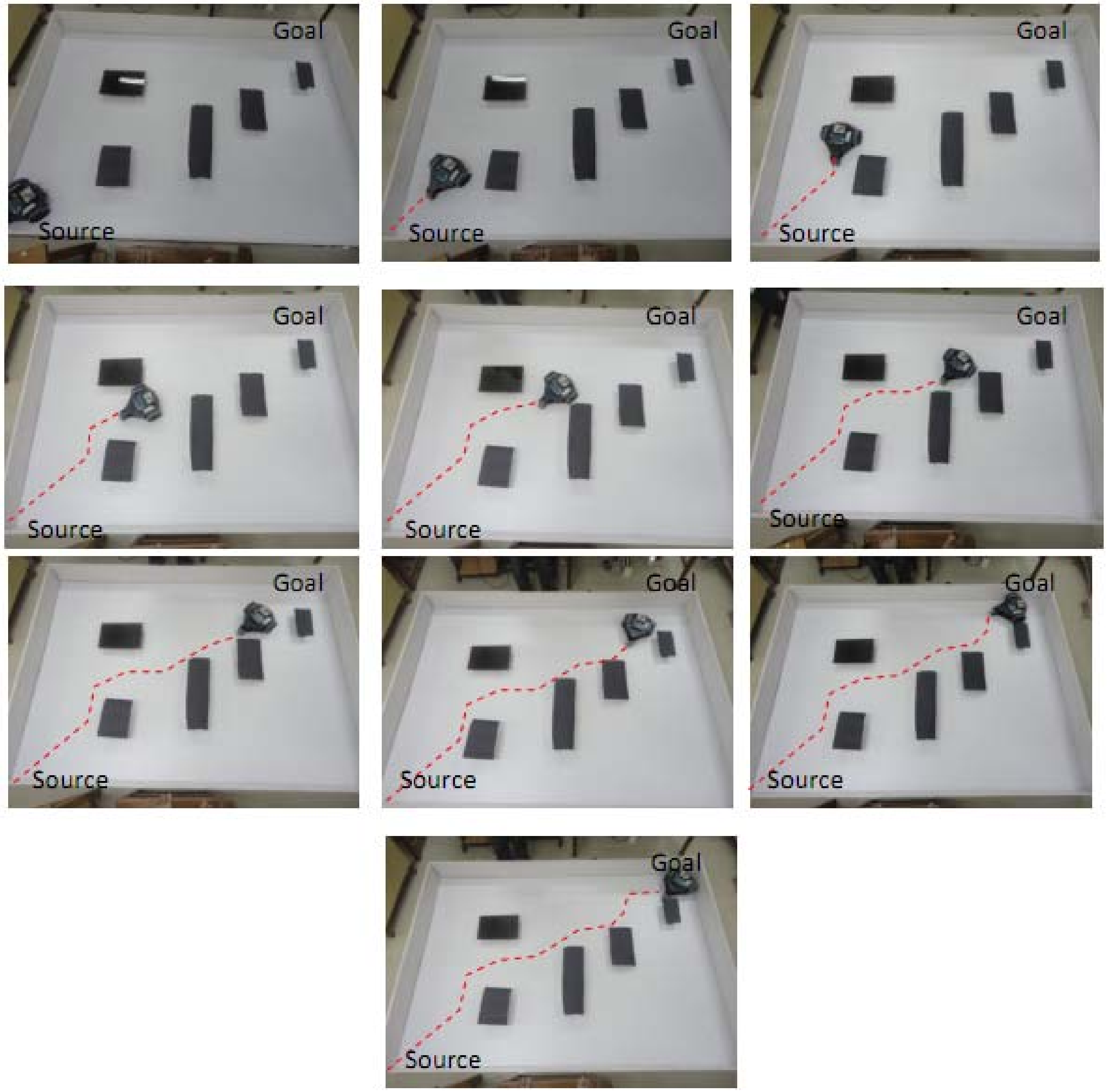


Fig. 9.8. Path taken by the robot during experiment using NN controller.

.

In chapter eight, hardware aspects of the Stingray robot has been given which is the robot used in all the experiments. Proper Hardware implementation of model mobile robot leads to the successful experimental verification of specified navigational algorithms.

Sl. No.	Algorithm used to find the robot's path	Path length in simulation achieved by proposed technique in cm.	Path length in experiment achieved by proposed technique in cm.	Error in %
1.	Fuzzy logic controller	277.7	278.0	0.108
2.	ABC Algorithm	249.3	250.0	0.280
3.	Type 2 fuzzy logic	258.2	258.33	0.050
4.	Backpropagation algorithm NN	252.0	253.0	0.397

Table: 9.1. Showing the Comparison between simulation and experimental results

Scale for simulation: 1 = 27.777

The table 9.1 shows the comparision between various algorithms used and their outputs in terms of the path length. The result also shows the error in percentage for each controller regarding the difference in length covered by the robot in simulation and experiment.

9.1 DISCUSSION

The fuzzy logic technique is supposed to mimic the human decision making power but it does not guarantee an optimal path generation. However, the error in the simulation and experimental analysis is low. The path generated by the robot is not an optimal one but it is generated depending upon the logic it works accordingly. The table 4.2 gives some of the values for the logical inputs and outputs. However, fuzzy logic also leaves some ambiguities in its operation. This is overcome by the type 2 fuzzy logic controller. The results found through type 2 fuzzy logic controller is found to be more satisfactory than fuzzy logic controller. The figure 5.5 shows the path taken by the robot using a type two fuzzy controller and it is better than the fuzzy controller.

Further the neural network technique is a learning mechanism and the accuracy is more for more number of epochs. The result of lesser number of epochs is more error. The results shown in fig 7.3 show the accuracy of the neural controller. The neural network designed for this robot is shown in fig 7.1. It shows all the inputs and outputs of the network structure. Artificial bee colony algorithm is a biologically inspired technique. The fact that the honey bees search for their food and then use the optimal path to reach it cannot be denied. Their method of finding their food is shown in chapter 6. The mathematical operations done to find this method can be viewed in this chapter. The fig 6.3 shows the path of the robot that undertakes to reach its goal by using the bee colony algorithm.

9.2 CONCLUSION

The results show that the while navigating using fuzzy controller the mobile takes a longer but safer path to reach the destination. As the type 2 controller is more accurate than the fuzzy logic controller hence the path taken by the mobile robot using the type 2 fuzzy controller is a bit shorter. Similarly, the backpropagation algorithm, being a learning algorithm uses a shorter path than the fuzzy controllers. However, the artificial bee colony algorithm is an evolutionary algorithm and when the mobile robot uses this algorithm the length of the path taken from the source to goal becomes even shorter. In fact the bee colony algorithm gives the shortest path length amongst all the three algorithms used.

CHAPTER 10

CONCLUSION AND FUTURE WORK

10. CONCLUSION AND FUTURE WORK

The previous chapters have presented the background, approach and results of this research in a detailed way. This chapter summarises the conclusions of the research and proposes idea for future work. It recapitulates the main contributions, conclusions of the present investigations and space for additional works. This investigation anticipates for making the following contributions to the domain of navigational path analysis of mobile robots in diverse environments. The major intents of this research work have been to find out efficient control techniques for mobile robot navigation in crowded real world situations by avoiding collision with obstacles arranged in a chaotic way.

10.1 CONCLUSIONS

In the present research, attempt has been made to solve a problem related to navigational path analysis of mobile robots in various environments. The investigation was carried out in several stages as follows:

1. From the kinematic analysis of mobile robot left wheel and right wheel velocities of the mobile robot has been calculated. From the wheel velocities, steering angle for the robot is calculated

2. Next the fuzzy controller uses the human decision making concept to find a path for the robot to reach from its source to its destination.

3. The fuzzy technique is further magnified and improved using the type 2 fuzzy logic. The uncertainties are minimized and accuracy is maximised.

4. Further, an evolutionary technique called the bee algorithm inhibits the robot with the navigational accuracies of honey bees which they use to find their food and reach their food source.

5. The neural controller is yet another technique used to teach the robot to move around in its environment by avoiding collisions and reaching the goal safely at its goal.

6. The hardware was described for the Sting Ray robot which has been used as the robot to navigate in experimental simulation.

Performance measure has been carried out through the comparison between simulation and experimental results for different environmental scenarios in terms of path travelled and they are found to be good agreement

This research is committed to appraise the performances of controllers during navigation of mobile robot in different simulation and experimental environmental scenarios along with comparison with previous research work for endorsement.

10.2 FUTURE WORKS

This work provides a foundation for future expansion of integrated designing approaches of intelligent controller based on artificial intelligence technique. Autonomous navigation in various environments is still an open area of research. There are a number of interesting directions to pursue as future work. The suggestions with several crucial and promising researches for future investigation are as follow. In the current research work, the techniques developed for navigational path analysis of mobile robot enable the robots to avoid collision among each other and with static obstacles. The current effort affords a base for forthcoming growth of cohesive designing approaches of sensible controller based on artificial intelligence technique enriched with human perception. Regardless of all research that has been conducted, autonomous navigation in various environments is still a vast area of research.

However, further development of the techniques may be required for the avoidance of moving obstacles other than the robots. This will make the algorithm more effective in dealing with unpredictable real life situations. The navigational techniques developed in this research work are capable of detecting and reaching the static targets. Further modifications in these navigational techniques may be carried out so that the robots can not only detect dynamic targets but also reach them using an optimum path. Further research is required for cooperative behaviour coordination between the robots for task and handling a particular object by avoiding static as well as moving obstacles.

REFERENCES

1. Patrick F.M. and Neuman C.P., Kinematic modelling of wheeled mobile robots, *Journal of Robotic Systems*, 4(2), 1986, 281 – 340.

2. Jones J. L. and Flynn A.M., Mobile robots: Inspiration to implementation, *A K Peters,Wellesley*, MA, 1993.

3. Alexander J.C, and Maddocks J. H., On the kinematics of wheeled mobile robots. *International Journal of Robotics Research*, 8(5), 1989, 15–27.

4. Fierro R. and Lewis F.L., Control of a nonholonomic mobile robot using neural networks,*IEEE Transaction on Neural Networks*, 9, 1998, 589–600.

5. Neimark J. I. and Fufaev F. A., Dynamics of nonholonomic systems, *American Mathematical Society, Providence*, RI, 1972.

6. Muir P.F., and Neuman C.P., Kinematic Modeling of Wheeled Mobile Robots, *Journal of Robotic Systems*, 1987, 4(2), pp. 281 – 340.

7. Batlle J.A. and Barjau A., Holonomy in mobile robots, *Robotics and Autonomous Systems*, 57 (2009) 433–440.

8. Ailon A. and Zohar I., Point-to-point control and trajectory tracking in wheeled mobile robots: some further results and applications, *in: Proc. Intentional Federation Automatic Control Conf., Seoul*, *Korea*, 2008.

9. Alexander J.C., and Maddocks J.H., On the Kinematics of Wheeled Mobile Robots, *The International Journal of Robotics Research*, 1989, Vol. 8(5), pp. 15-27.

10. Chakraborty Nilanjan and Ghosal Ashitava, ―Kinematics of wheeled mobile robots on uneven terrain, *Mechanism and Machine Theory* 39 (2004) 1273–1287.

11. Gobi Adam F. and Pedrycz Witold, The potential of fuzzy neural networks in the realization of approximate reasoning engines, *Fuzzy Sets and Systems* 157 (2006) 2954 – 2973.

12. Borenstein J., Everett H.R., Feng L. and Wehe D., Mobile robot positioning Sensors and techniques, *Journal of Robotics Systems* 14 (4) (1997) 231–249.

13. Moon Jong-Woo, Park Chong-Kug and Harashima Fumio,―Kinematic Correction of a Differential Drive Mobile Robot and a Design for Velocity Trajectory with Acceleration Constraints on Motor Controllers, *In Proc. IEEE International Conference on Intelligent Robots and Systems*, 1999, pp. 930-935.

14. Bloch A.M., Reyhanoglu M. and McClamorch H., Control and stabilization of nonholonomic dynamic systems, *IEEE Tran. on Automatic Control*, 37, 11, pp. 1746-1757, (1992).

15. Fierro R. and Lewis L., Control of a nonholonomic mobile robot: back stepping kinematics into dynamics, *proc. of the 34th IEEE Conf. on Decision & Control, New Orleans*, pp. 3805-3810, (1995).

16. Jin Yaochu and Jiang Jinping, Techniques in Neural-network-based Fuzzy System Identification and Their Application to control of complex systems□, Fuzzy Theory Systems Techniques and Applications, 1 (1999) 112-128.

17. Tanner H.G. and Kyriakopoulos K.J., Discontinuous back stepping for stabilization of nonholonomic mobile robots, *proc. of IEEE Conf. on Robotics and Automation, Washington DC*, pp. 3948-3953, (2002).

18. Chung Yongoug, Park Chongkug and Harashima Fumio, A Position Control Differential Drive Wheeled Mobile Robot, *IEEE Transactions on Industrial Electronics*, Vol. 48, No. 4, August 2001, pp.853-863

19. Hou Zeng-Guang, Zou An-Min, Cheng Long, and Tan Min, Adaptive Control of an Electrically Driven Nonholonomic Mobile Robot via Backstepping and Fuzzy Approach, *IEEE Transactions on Control Systems Technology*, Vol. 17, No. 4, July 2009, pp. 803-815.

20. Pourboghrat Farzad and Karlsson Mattias P., Adaptive control of dynamic mobile robots with nonholonomic constraints, *Computers and Electrical Engineering* 28 (2002) 241–253.

21. Martins Felipe N., Celeste Wanderley C., Carelli Ricardo, Ma´ rio Sarcinelli-Filho and Bastos-Filho Teodiano F., An adaptive dynamic controller for autonomous mobile robot trajectory tracking, *Control Engineering Practice* 16 (2008) 1354– 1363.

22. Deng M., Inoue A., Sekiguchi K. and Jiang L., Two-wheeled mobile robot motion control in dynamic environments, *Robotics and Computer-Integrated Manufacturing* 26 (2010) 268–272.

23. Dixon W. E., Jiang Z. P. and Dawson D. M., Global exponential set point control of wheeled mobile robots: a Lyapunov approach, *Automatica 2000*, Vol. 36, pp. 1741-1746.

24. Ge SS and Cui YJ, Dynamic motion planning for mobile robots using potential field method, *Autonomous Robots 2002*; 13: 207–22.

25. Pavlovskii V.E., Evgrafov V.V., Pavlovskii V.V. and Petrovskaya N.V., ―Mobile Robots with Two Independent Coaxial Drive Wheels: Dynamics and the Schemes of Control, *published in Vestnik Moskovskogo Universiteta, Matematika. Mekhanika*, 2009, Vol. 64, No. 1, pp. 44–50.

26. Tsuchiya Kazuo, Urakubo Takateru and Tsujita Katsuyoshi, A Motion Control of a Two-Wheeled Mobile Robot, 1999 *IEEE*.

27. Arvin Farshad, Samsudin Khairulmizam, and Nasseri M. Ali, Design of a Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation, *IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA 2009) Monash University, Sunway campus, Malaysia*, 25th & 26th July 2009, pp.143-147.

28. Chakraborty Nilanjan and Ghosal Ashitava, Kinematics of wheeled mobile robots on uneven terrain, *Mechanism and Machine Theory* 39 (2004) 1273–1287.

29. Zhang Huai-Xiang, Dai Guo-Jun and Zeng Hong, A Trajectory Tracking Control Method for Nonholonomic Mobile Robots, *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China*, 2-4 Nov. 2007, pp. 7-11.

30. Do K. D., Jiang Z. P. and Pan J., Simultaneous Tracking and Stabilization of Mobile Robots: An Adaptive Approach, *IEEE Transactions on Automatic Control*, Vol. 49, No. 7, July 2004, pp.1147-1152.

31. Parhi Dayal R., Navigation of Mobile Robots Using a Fuzzy Logic Controller□, *Journal of Intelligent and Robotic Systems*, Volume 42, Number 3 / March, 2005.

32. Zohar Ilan, Ailon Amit Ailon and Rabinovici Raul, Mobile robot characterized by dynamic and kinematic equations and actuator dynamics: Trajectory tracking and related application, *Robotics and Autonomous Systems*, 59 (2011) 343–353.

33. Ailon A. and Zohar I., Control strategies for driving a group of nonholonomic kinematic mobile robots in formation along a time-parameterized path, *to be published in IEEE/ASME Transactions on Mechatronics, (posted on IEEE Xplore)* 2011.

34. Fliess M., Lvine J., Martin P. and Rouchon P., Flatness and defect of non-linear systems: introductory theory and examples, *International Journal of Control*, 61 (1995) 1327–1361.

35. Fierro R. and Lewis L., Control of a nonholonomic mobile robot: backstepping kinematics into dynamics, *proc. of the 34th IEEE Conf. on Decision & Control, New Orleans*, pp. 3805-3810, (1995).

36. Gandhi P.S. and Ghorbel F., High-speed precision tracking with harmonic drive systems using integral manifold control design, *International Journal of Control*, 78(2), 2005, 112– 121.

37. Pathak P.M., Mukherjee A. and Dasgupta A., Interaction torque control by impedance control of space robots, *Simulation, 85(7)*, 2009, 451-459.

38. Han Soonshin, Choi ByoungSuk, Lee JangMyung, A precise curved motion planning for a differential driving mobile robot, *Mechatronics 18* (2008) 486–494.

39. Gracia Luis and Tornero Josep, Characterization of zero tracking error references in the kinematic control of wheeled mobile robots, *Robotics and Autonomous Systems 57* (2009) 565_577.

40. Vandorpe J., and Van Brusell H., A Reflexive Navigation Algorithm for an Autonomous Mobile Robot, *International Conference on Multisensor Fusion and Integration for Intelligent System, Las Vegas, NV*, 1994, pp. 251 - 258.
41. Hoffmann F., Evolutionary algorithms for fuzzy control system design, Proceedings ofthe IEEE, 89(9), 2001, 1318-1333.
42. Zadeh L., Fuzzy sets, *Information and Control*, 8(3), 1965, 338-353.
43. Xu W. L., and Tso S. K., Sensor-based fuzzy reactive navigation of a mobile robotthrough local target switching, *IEEE Transactions on Systems, Man, And Cybernetics-Part C, Application and reviews, 29(3*), 1999, 451-459.
44. Teodorescu H.N.L., Chelaru M., Kandel A. Tofan I. and Erimia M., Fuzzy methods in tremor assessment, prediction and rehabilitation. *Artificial Intelligence in Medicine*,21(1), 2001, 107-130.
45. Mamdani E.H. and Assilian S., An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies, 7*, 1975, 1-13.
46. Guo S., Peters L. and Surmann H., Design and application of an analog fuzzy logiccontroller, *IEEE Transactions on Fuzzy Systems, 4(4),* 1996, 429-438.
47. Sousa J.M.C. and Kaymak U., Fuzzy decision making in modeling and control, *World Scientific Series In Robotics And Intelligent Systems, 27, World Scientific Publishing, India*, 2002.
48. Huq R., Mann G.K.I. and Gosine R.G., Behaviour-modulation technique in mobilerobotics using fuzzy discrete event system, *IEEE Transactions on Robotics, 22(5),* 2006,903-916.
49. Li H. and Yang S. X., A Behaviour-based mobile robot with a visual landmark recognition system, *IEEE/ASME Transactions on Mechatronics, 8(3),* 2003, 390-400
50. Galichet S. and Foulloy L., Fuzzy controllers: Synthesis and equivalences, *IEEETransactions on Fuzzy Systems, 3(2),* 1995,140-148.

51. Juang, C.-F. and Hsu C.-H., Reinforcement ant optimised fuzzy controller for mobilerobot wall following control, *IEEE Transactions on Industrial Electronics, 56,* 2009,

52. Cuesta F., Gordillo F., Aracil J. and Ollero A., Stability analysis of nonlinearmultivariable takagi–sugeno fuzzy control systems, *IEEE Transactions on Fuzzy Systems*,7(5), 1999, 508-520.

53. Cupertino F., Giordano V., Naso D. and Delfine L., Fuzzy control of a mobile robotics, *IEEE Robotics and Automation Magazine*, 74-81.

54. Yang S. X., Li H., Meng M.Q.-H. and Liu P.X., An embedded fuzzy controller for a behaviour-based mobile robot with guaranteed performance, *IEEE Transactions on Fuzzy Systems* , 12(4), 2004, 436-446.

55. Sim K.B., Byun K.S. and Harashima F., Internet-based teleoperation of an intelligent robot with optimal two-layer fuzzy controller, *IEEE Transactions on Industrial Electronics*, 53(4), 2006, 1362-1372.

56. Chen X., Watanabe K., Kiguchi K., and Izumi K., An ART-based fuzzy controller for the adaptive navigation of a quadruped robot, *IEEE/ASME Transactions on Mechatronics*, 7(3), 2002, 318-328.

57. Li Z. and Xu C., Adaptive fuzzy logic control of dynamic balance and motion for wheeled inverted pendulums, *Fuzzy Sets and Systems*, 160, 2009, 1787–1803.

58. Lin H.H., Tsai C.C., and Hsu J.C., Ultrasonic localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering, *IEEE* Transactions on Industrial Electronics, 57(9),2008, 2024-2034.

59. Naranjo J.E., González E., García R., Pedro T.D., and Haber R.E., Power-steering control architecture for automatic driving, *IEEE Transactions on Intelligent transportation Systems*, 6(4), 2005, 406-415.

60. Lilly J.H., Evolution of a negative-rule fuzzy obstacle avoidance controller for an autonomous vehicle, *IEEE Transactions on Fuzzy Systems*, 15(4), 2007,718-728.

61. Ye C., Yung N.H. C. and Wang D., A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance, *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 33(1), 2003, 17-27.

62. Martinez A., Tunstel E. and Jamshidi M., Fuzzy-logic based collision-avoidance for a mobile robot, *Robotica*, 1994, 12(6), 521-527.

63. Boem H.R. and Cho H.S., A sensor-based navigation for a mobile robot using fuzzy-logic and reinforcement learning, *IEEE transactions on systems man and cybernetics Part B*, 25(3), 1995, 464-477.

64. Chiu C.S. and Lian K.Y. and Liu P., Fuzzy gain scheduling for parallel parking a car-Like robot, *IEEE Transactions on Control Systems Technology*, 13(6), 2005, 1084-1092.

65. Baturone I., Moreno-Velo F.J., Sánchez-Solano S., and Ollero A., Automatic design of fuzzy controllers for car-like autonomous robots, *IEEE Transactions on Fuzzy Systems*, 12(4), 2004, 447-465.

66. Hwang C.L., Chang L.J. and Yu Y.S., Network-based fuzzy decentralized sliding-mode control for car-like mobile robots, *IEEE Transactions on Industrial Electronics*, 54(1), 2007, 574-585.

67. Wang L.X., Modelling and control of hierarchical systems with fuzzy systems, *Automatica*, 33(6), 1997, 1041-1053.

68. Seraji H. and Howard A., Behaviour-based robot navigation on challenging terrain: A fuzzy logic approach, *IEEE Transactions on Robotics and Automation, 18(3)*, 2002, 308-321.

69. Hagras H., Callaghan V. and Colley M., Outdoor mobile robot learning and adoption, IEEE Robotics and Automation Magazine, 2001, 53-69.

70. Antonelli G., Chiaverini S. and Fusco G., A Fuzzy-logic-based approach for mobile robot path tracking, *IEEE Transactions on Fuzzy Systems*, 15(2), 2007, 211-221.

71. Hoffmann F., Schauten D. and Hölemann S., Incremental Evolutionary Design of TSK Fuzzy Controllers, *Transactions on Fuzzy Systems*, 15(4),2007, 563-577.

72. Moulton K.M, Cornell A. and Petriu E., A fuzzy error correction control system, *IEEE Transactions on Industrial Electronics*, 50 (5), 2001, 1456-1463.

73. Das T. and Kar I.N., Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots, *IEEE Transactions on Control Systems Technology,*14(3), 2006, 501-510.

74. Chiou J.S. and Wang K.Y., Application of a hybrid controller to a mobile robot, *Simulation Modelling Practice and Theory, 16*, 2008, 783–795.

75. Yang X., Moallem M. and Patel R. V., A Layered goal-oriented fuzzy motion planning strategy for mobile robot navigation, *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 35(6), 2005, 1214- 1224.

76. Kim S.H., Park C. and Harashima F., A self-organized fuzzy controller for wheeled mobile robot using an evolutionary algorithm, *IEEE Transactions on Industrial Electronics*, 48(2), 2001, 467-474.

77. Abdessemed F., Benmahammed K. and Monacelli E., A fuzzy-based reactive controller for a non-holonomic mobile robot, *Robotics and Autonomous Systems*, 47, 2004, 31–46.

78. Luminada B.I., Moreno-Velo F.J., Blanco V., and Ferruz, J., Design of embedded DSP based fuzzy controllers for autonomous mobile robots, *IEEE Transactions on Industrial Electronics*, 55(2), 2008, 928-936.

79. Gueaieb W., and Miah M.S., An Intelligent mobile robot navigation technique using RFID technology*, IEEE Transactions on Instrumentation and Measurement, 57(9),* 2008,1908-1917.

80. Lee T.H., Lam H.K., Leung F.H.F. and Tam P.K.S., A practical fuzzy logic controller for the path tracking of wheeled mobile robots, *IEEE Control Systems Magazine*, 2003, 60-65.

81. Sánchez-Solano S., Cabrera A.J., Baturone I., Moreno-Velo F.J., and Brox M., FPGA implementation of embedded fuzzy controllers for robotic applications, *IEEE* Transactions on Industrial Electronics, 54(4), 2007, 1937-1945.

82. Nefti S., Oussalah M., and Kaymak U., A new fuzzy set merging technique using inclusion based fuzzy clustering, *IEEE Transactions on Fuzzy Systems*, 16(1), 2008, 145-161.

83. Pallottino L., Scordio V. G, Bicchi A. and Frazzoli E., Decentralized cooperative policyfor conflict resolution in multivehicle systems, *IEEE Transactions on robotics*, 23(6), 2007, 1170-1183.

84. Krieger M.J.B., Billeter J.B. and Laurent Keller L., Ant-like task allocation and recruitment in cooperative robots, *Nature*, 406, 2000, 992-995.

85. Abdi, H., Valentin, D., & Edelman, B. (1999). *Neural networks. Thousand Oaks* (CA): Sage.

86. Abdi, H., Valentin, D., Edelman, B., O’Toole. A.J. (1996). A Widrow-Hoff learning rule for a generalization of the linear auto-associator. *Journal of Mathematical Psychology*, 40, 175–182.

87. Bishop, C. M. (1995) *Neural networks for pattern recognition.* Oxford, UK: Oxford University Press.

88. Cherkassky, V., & Mulier, F. (1998). *Learning from data*. New York: Wiley

89. Parhi D.R. and Singh M.K. Real time navigational control of mobile robots using an artificial neural network. Journal of Mechanical Engineering Science part C, 223(7), 2009, 1713-1725.

90. McCulloch W. S. and Pitts W., A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics, 5*, 1943. 115-133.

91. Hebb, D.O., *The organisation of behaviours*, John Wiley & Sons, New York, 1949.

92. Suzuki M., Floreano D. and Paolo E.A.D., The contribution of active body movement to visual development in evolutionary robots, *Neural Networks*, 18, 2005, 656–665.

93. Rosenblatt, F., *Principles of neurocomputing*, Addison-Wesley Publishing Co., 1959.

94. Widrow, B. and Hoff. M.E., *Adaptive switching circuits*, IRE Westcon Convention
    a. Record, Part 4, 1960, 96-104.

95. Minsky M. and Papert S. *Perceptrons: An introduction to computational geometry*, The MIT Press, Cambridge, MA, 1969.

96. Duda, R., Hart, P.E., Stork, D.G. (2001) *Pattern classification.* New York: Wiley.

97. Hastie T., Tibshirani R., Friedman J. (2001). *The elements of statistical learning*. New-Yrok: Springer-Verlag.

98. Ijspeert A.J., Central pattern generators for locomotion control in animals and robots: A review, *Neural Networks*, 21, 2008, 642–653.

99. Werbos P.J., Intelligence in the brain: A theory of how it works and how to build it, *Neural Networks*, 22, 2009, 200-212.

100. Tani J., Nishimoto R., and Paine R.W., Achieving “organic compositionality” through self-organization: Reviews on brain-inspired robotics experiments, *Neural Networks*, 21,2008, 584–603.

101. Webb B., Robots in invertebrate neuroscience, *Nature*, 417, 2002, 359-363.

102. Yoo S. J., Choi Y.H. and Park J.B., Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: Adaptive learning rates approach, *IEEE Transactions on Circuits and Systems*, 153(6), 2006, 1381-1394.

103. Braganza D., Dawson D. M., Walker I. D., and Nath N., A neural network controller for continuum robots, *IEEE Transactions on Robotics*, 23(5), 2007, 1270-1277.

104. Gu D. and Hu H., Neural predictive control for a car-like mobile robot, *International* Journal of Robotics and Autonomous Systems, 39(2-3), 2002, 1-15.

105. Kondo T., Evolutionary design and behaviour analysis of neuromodulatory neural networks for mobile robots control, *Applied Soft Computing*, 7, 2007, 189–202.

106. Smagt P.V., Cerebellar control of robot arms, *Journal of Connection Science*, 10, 1998, 301–320.

107. Scheiera C., Pfeifera R. and Kunyioshi Y., Embedded neural networks: exploiting constraints, *Neural Networks*, 11, 1998, 1551–1569.

108. Brooks R.A., Intelligence without representation, *Artificial Intelligence*, 47, 1991, 139–160.

109. Clark A. and Thornton C., Trading spaces: computation, representation and the limits of uninformed learning, *Behavioural and Brain Sciences*, 20, 1997, 57–90.

110. Capi G., Doya K., Evolution of recurrent neural controllers using an extended parallel genetic algorithm, *Robotics and Autonomous Systems*, 52, 2005, 148–159.

111. Hu¨lse M., Wischmann S. and Pasemann F., Structure and function of evolved neurocontrollers for autonomous robots, *Connection Science*, 16(4), 2004, 249–266.

112. Arena P., De Fiore S., and Patané L. Cellular Nonlinear Networks for the emergence of perceptual states: Application to robot navigation control, *Neural Networks*, 2009, *In* Press. perceptual states: *Application to robot navigation control*, *Neural Networks*, 2009, *In*Press.

113. Kim M.Y. and Cho H., Three-dimensional map building for mobile robot navigation environments using a self-organizing neural network, *Journal of Robotic Systems*, 21(6), 2004, 323–343.

114. Ripley, B.D. (1996) *Pattern recognition and neural networks*. Cambridge,MA: Cambridge University Press.

115. Sch¨olkopf B., Smola, A.J. (2003). *learning with kernels.* Cambridge (MA):MIT Press.

116. Vapnik, V. N. (1999) *Statistical learning theory*. New York: Wiley.

117. Rumelhart, D. and J. McClelland (1986). *Parallel Distributed Processing*. MIT Press, Cambridge,Mass

118. Troncoso J.M.C., Sa´nchez J.R.A. and Lo´pez F.d.l.P., Discretized ISO-learning neural network for obstacle avoidance in reactive robot controllers, *Neurocomputing*, 72, 2009, 861–870.

119. Jaksa R., Sincak P. and Majernik P., Back propagation in supervised and reinforcement learning for mobile robot control, *International Conference on Computational Intelligence for Modeling, Control and Automation, Vienna*, 1999, 1-7.

120. Maravall D., de Lope J. and Martı´n J. A. H., Hybridizing evolutionary computation and reinforcement learning for the design of almost universal controllers for autonomous robots, *Neurocomputing*, 72, 2009, 887–894.

121. Yong Duan Y., Liu Q. and Xu X.H., Application of reinforcement learning in robot soccer, *Engineering Applications of Artificial Intelligence, 20*, 2007, 936–950.

122. Nelson A.L., Grant E. and Henderson T.C., Evolution of neural controllers for competitive game playing with teams of mobile robots, *Robotics and Autonomous Systems*, 46(3), 2004, 135–150.

123. Memon Q. and Khan S., Camera calibration and three-dimensional world reconstruction of stereo-vision using neural networks, *International Journal of Systems Science*, 32(9), 2001, 1155- 1159.

124. Liatsis P., Goulermas J.Y., Zeng X.-J. and Milonidis E., A flexible visual inspection system based on neural networks, *International Journal of Systems Science*, 40(2), 2009, 173–186.

125. Trentin E. and Cattoni R., Learning perception for indoor robot navigation with a hybrid hidden markov model/recurrent neural networks approach, *Connection Science*, 11(3-4), 1999, 243- 265.

126. Iglesias R., Nehmzow U. and Billings S.A., Model identification and model analysis in robot training, *Robotics and Autonomous Systems*, 56, 2008, 1061-1067.

127. Murray J.C., Erwin H.R. and Wermter S., Robotic sound-source localisation architecture using cross-correlation and recurrent neural networks, *Neural Networks*, 22, 2009, 173-189.

128. Burgsteiner H., Imitation learning with spiking neural networks and real-world devices, *Engineering Applications of Artificial Intelligence*, 19, 2006, 741–752.

129. Cheng T., Lewis F.L. and Abu-Khalaf M., Aneural network solution for fixed-final time optimal control of nonlinear systems, *Automatica*, 43, 2007, 482 – 490.

130. Maguire L.P., McGinnity T.M., Glackin B., Ghani A., Belatreche A. and Harkin J., Challenges for large-scale implementations of spiking neural networks on FPGAs, *Neurocomputing*, 71, 2007, 13–29.

131. Wanga X., Houa Z.G., Zoua A., Tana M. and Chenga L., A behaviour controller based on spiking neural networks for mobile robots, *Neurocomputing*, 71, 2008, 655–666.

132. Sato M. and Ishii K., A neural network based controller for a wheel type mobile robot, *International Congress Series*, 1291, 2006, 261– 264.

133. Sato M., Kanda A. and Ishii K., Performance evaluation of a neural network controller system for a wheel type mobile robot, *International Congress Series*, 1301, 2007, 160–163.

134. Verschure P.F.M.J., Voegtlin T. and Douglas R.J., Environmentally mediated synergy between perception and behaviour in mobile robots, *Nature*, 425(9), 2003, 620-624.

135. Ca˜nas J.M. and Vicente Matell´an V., From bio-inspired vs. psycho-inspired to ethoinspired robots I, *Robotics and Autonomous Systems*, 55, 2007, 841–850.

136. Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Springer, Berlin.

137. Young, D. *Formal Computational Skills* Course Notes
138. Bar-Yam, Y. (1997). *Dynamics of Complex Systems*. Addison-Wesley.
139. Kauffman, S. (1993) *Origins of Order*, Oxford University Press.

140. McCulloch, W. and W. Pitts (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity.*Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133.

141. Zadeh, L. A. (1965). "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353

142. E. Hisdal, The IF THEN ELSE statement and interval-valued fuzzy sets of higher type, *Internat. J. Man-Mach. Stud*. 15 (1981) 385–455.

143. R.I. John, Type 2 fuzzy sets: an appraisal of theory and applications, Internat. J. Uncertainty, *Fuzziness Knowledge-Based Systems 6 (6)* (1998) 563–576.

144. S. Mabuchi, An interpretation of membership functions and the properties of general probabilistic operators as fuzzy set operators .II. Extension to three-valued and interval-valued fuzzy sets, *Fuzzy Sets and Systems 9*2 (1) (1997) 31–50.

145. D.G. Schwartz, The case for an interval-based representation of linguistic truth, *Fuzzy Sets and Systems 17* (1985) 153 165.

146. Karnik N. N. and Mendel, J. M. (2001). "Centroid of a type-2 fuzzy set," *Information Sciences*, vol. 132, pp. 195-220.

147. J.M. Mendel, Fuzzy logic systems for engineering: a tutorial, Proc. *IEEE 83 (3)* (1995) 345–377.
148. Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, *Cardiff University*, UK, 2005

149. Duc Truong Pham, Ashraf Afify, EbubekirKoc, Manufacturing cell formation using the Bees Algorithm . *IPROMS 2007 Innovative Production Machines and Systems Virtual Conference*, Cardiff, UK.

150. D. T. Pham, E. Koç, J. Y. Lee, and J. Phrueksanant, Using the Bees Algorithm to schedule jobs for a machine, Proc *Eighth International Conference on Laser Metrology, CMM and Machine Tool Performance, LAMDAMAP*, Euspen, UK, Cardiff, p. 430–439, 2007.

151. Pham D.T., Ghanbarzadeh A., Koç E., Otri S., Rahim S., and M.Zaidi "*The Bees Algorithm – A Novel Tool for Complex Optimisation Problems* "", Proceedings of IPROMS 2006 Conference , pp.454–461

152. Von Frisch K. Bees: Their Vision, Chemical Senses and Language. (Revised edn) *Cornell University Press*, N.Y., Ithaca, 1976.

153. Zadeh L.A., A new direction in AI—Toward a computational theory of perceptions, *AIMagazine*, 22(1), 2001, 73–84.

154. Parhi D R and Singh M K, ―Intelligent fuzzy interface technique for the control of an autonomous mobile robot, Proceedings of the Institution of Mechanical Engineers, Part C: *Journal of Mechanical Engineering Science*, Volume 222, Number 11 / 2008.

155. Herrero-Pérez D., Martínez-Barberá H., LeBlanc K. and Saffiotti A., ―Fuzzy uncertainty modeling for grid based localization of mobile robots, International Journal of Approximate Reasoning 51 (2010) 912–932.

156. Qing-yong BAO, Shun-ming LI, Wei-yan SHANG and Mu-jin AN, ―A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments, *International Conference on Artificial Intelligence and Computational Intelligence*, 2009, vol.2, pp.257-261.2009

157. Singh M.K., Parhi D.R., Bhowmik,S., Singh P.K.; Swarming of multiple mobile robots for searching operation using ACO, *International Conference on ICSCIS-07*, December 27-29,2007, JEC Jabalpur, India

158. Zhu A. and Yang S. X., Neuro-fuzzy based approach to mobile robot navigation in unknown environments, *IEEE Transactions on Systems, man, and cybernetics part C*, 37(4), 2007, 610-621.

159. Arkin R. C. Motor schema-based mobile robot navigation, *International Journal of Robotics Research*, 8(4), 1989, 92–112.

160. L. A. Zadeh, “The Concept of a Linguistic Variable and Its Application to Approximate Reasoning–” *Information Sciences*, vol. 8, pp. 199-249,

161. Mendel, J. (2007). Advances In Type-2 Fuzzy Sets and Systems, *Information Sciences*, 177,1,pp. 84-110.

162. Hagras, H. (2007). Type-2 FLC’s: a New Generation of Fuzzy Controllers, *IEEE Computational Intelligence Magazine*, February 2007, pp. 30-43.

163. Mendel, J. (2001). Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice Hall, New Jersey.

164. John, R. & Coupland, S. (2007). Type-2 Fuzzy Logic A Historical View*, IEEE Computational Intelligence Magazine*, 2,1, pp. 57-62.

165. Karnik, N. & Mendel, J. (2001). Centroid of a Type-2 Fuzzy Set*, Information Sciences*, 132, pp.195-220.

166. Castro, J.; Castillo, O. & Melin, P. (2007). An Interval Type-2 Fuzzy Logic Toolbox for Control Applications, *Proceedings of the 2007 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2007,* London, UK.

167. Melgarejo, M.; Garcia, A. & Pena-Reyes, C. (2004). Computational Model and architectural proposal for hardware Type-2 Fuzzy System, *Proceedings of the 2nd IASTED conference on Neural Networks and Computational Intelligence*, Grindewald, Switzerland.

168. Melgarejo, M. & Pena-Reyes, C. A. (2007). Implementing Interval Type-2 Fuzzy Processors, *IEEE Computational Intelligence Magazine*, 2,1, pp. 63-71.

169. Torres, P. & Saez, D. (2008). Type-2 fuzzy logic identification applied to the modelling of a robot hand, *Proceedings of the 2008 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008*.

170. Figueroa, J.; Posada, J.; Soriano, J.; Melgarejo, M. & Rojas, S. (2005). A Type-2 Fuzzy Logic Controller For Tracking Mobile Objects In The Context Of Robotic Soccer Game, *Proceedings of the 2005 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE2005*.

171. Hagras, H. (2004). A Type-2 Fuzzy Logic Controller for Autonomous Mobile Robots, *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE2004*, Budapest, Hungary.

172. Martinez, R.; Castillo, O. & Aguilar, L. (2008). Optimization with Genetic Algorithms of Interval Type-2 Fuzzy Logic controllers for an autonomous wheeled mobile robot:A comparison under different kinds of perturbations, *Proceedings of the 2008 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008*.

173. Hagras, H. (2008). Developing a type-2 FLC through embedded type-1 FLCs*,* Proceedings of the2008 *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE* 2008, HongKong.

174. Leottau, L. &Melgarejo, M. (2010b). A Simple Approach for Designing a Type-2 Fuzzy Controller for a Mobile Robot Application, *Proceedings of the North American Fuzzy Information Processing Society's NAFIPS 2010*, Toronto, Canada, July 2010.

175. Melin, P. & Castillo, O. (2003). A new method for adaptive model-based control of non linearplants using type-2 fuzzy logic and neural Networks, *Proceedings of the 2003IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2003,* St. Louis, MO.

176. Wu, D. & Tan, W. (2004). A type-2 fuzzy logic controller for the liquid-level process, Proceedings of the 2004 *IEEE International Conference on Fuzzy Systems*, *FUZZIEEE2004.*

177. Leottau, L. &Melgarejo, M. (2010a). A Methodological Proposal for Implementing IntervalType-2 Fuzzy Processors over Digital Signal Controllers, *Journal of Applied Computer Science Methods*, v.2-1, June 2010, pp.61-81.

178. Seeley TD. The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies. Massachusetts: *Harvard University Press*, Cambridge, 1996.

179. Bonabeau E, Dorigo M, and Theraulaz G. Swarm Intelligence: from Natural to Artificial Systems. *Oxford University Press*, New York, 1999.

180. Camazine S, Deneubourg J, Franks NR, Sneyd J, Theraula G and Bonabeau E. Self-Organization in Biological Systems. Princeton: *Princeton University Press*, 2003.

181. D. T. Pham, E. Koç, A. Ghanbarzadeh, and S. Otri, Optimisation of the weights of multi-layered perceptrons using the Bees Algorithm, *Proc 5th International Symposium on Intelligent Manufacturing Systems*, Turkey, 2006.

182. D. T. Pham, A. Ghanbarzadeh, E. Koç, and S. Otri, Application of the Bees Algorithm to the training of radial basis function networks for control chart pattern recognition, Proc *5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (CIRP ICME '06)*, Ischia, Italy, 2006.

183. D. T. Pham, S. Otri, A. Ghanbarzadeh, and E. Koç, Application of the Bees Algorithm to the training of learning vector quantisation networks for control chart pattern recognition, *Proc Information and Communication Technologies (ICTTA'06),* Syria, p. 1624–1629, 2006.

184. D. T. Pham, A. J. Soroka, A. Ghanbarzadeh, E. Koç, S. Otri, and M. Packianather, Optimising neural networks for identification of wood defects using the Bees Algorithm, Proc 2006 *IEEE International Conference on Industrial Informatics*, Singapore, 2006.

LIST OF PUBLISHED AND ACCEPTED PAPERS

1. "Evolutionary algorithms and related techniques in the field of robotics: a review", Proceedings of International Conference on Advances in Modelling, Optimization and Computation(AMOC) 2011, Pg. No. 36-37, 6^{th} to 8^{th} December, 2011, Roorkee.
2. "Navigation of a Mobile Robot Using Fuzzy Logic", Proceedings of International Conference on Computer Science Engineering (ICCSE) 2012, Pg No. 128-130, 3^{rd}-4^{th} February 2012, Nagpur, Maharashtra.
3. "Fuzzy logic control of a WMR", Proceedings of IEEE sponsored International Conference on Computing, Communication and Application (ICCCA) 2012, Pg.No. 105-108, 26^{th}-29^{th} February 2012, Dindigul, Tamil Nadu.
4. "Cell decomposition: The way for path planning in complex environment", Proceedings of International Conference on Computing, Communication and Application (ICCCA) 2012, Pg.No. 138-142, 26^{th}-29^{th} February 2012, Dindigul, Tamil Nadu.
5. "Navigation of a Mobile Robot Using Fuzzy Logic" , Communicated to International Journal of Internet Computing(IJIC)..
6. "Analysis of Map Representation for Control of Mobile Robot in Various Environmental Situations", Communicated to Journal of Robotics.
7. "Brains in the artificial world", Communicated to International Journal of Artificial Intelligence and Computational Research(IJAICR)